MILANO 22-26 MAGGIO 1984

SOMMARIO

Direttore responsabile: Michele Di Pisa
Redattore capo: Alessandro De Simone
Foto di copertina: Franco Vignati
Impaginazione/illustrazioni: Francesco Amatori, Renato Caruso.
Hanno collaborato: Mauro Battisti, Giovanni Bellù, Maurizio Di Vizio, Marco Salvato, Danilo Toma.
Direzione, redazione: V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Mirco Croce (coordinatore), Paola Bevilacqua, Michela Prandini, Giorgio Ruffoni, Marco Ravagli, Roberto Sghirinzetti, Claudio Tidone, Villa Claudio - V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348/9/40
Prezzo e abbonamenti: Prezzo per una copia Lire 2.500
Arretrati il doppio. Abbonamento annuo (dieci fascicoli) Lire 22.000
Abbonamento annuo cumulativo alle riviste Computer e Commodore Computer Club (tariffa riservata agli studenti) L. 34.000. I versamenti vanno indirizzati a: Commodore Computer Club, mediante assegno bancario, vaglia o utilizzando il c/c postale n. 31532203.
Composizioni: Minisystems Italia
Selezioni: Org. A.G.
Stampa: Lito 3 - Cologno Monzese
Registrazione: Tribunale di Milano n. 2/10/1982 N.ro 370. Sped in abb. post. gr. III. Pubblicità inferiore al 70%

COMMODORE 64,

Fatti un regalo intelligente: un computer dalle caratteristiche eccezionali. Vediamole.

1. Commodore 64 è potente, sofisticato, professionale.

2. Ha un'incredibile memoria (64 K), un sintetizzatore sonoro professionale, produce effetti tridimensionali.

3. Ti diverti perchè è anche un sofisticato videogioco.

4. Con Commodore 64 entri nel futuro, tasto dopo tasto.

ORA CHE CE L'HAI...

5. Ora hai una gamma di programmi già pronti ancora più vasta: lo usi nel lavoro, a casa, a scuola, in tante applicazioni dall'hobby al professionale.

6. Commodore 64 oggi lo puoi avere a prezzo davvero speciale: approfittane però perchè sta andando a ruba, e chi primo arriva...

commodore
COMPUTER

DOMANDE RISPOSTE

DOMANDE RISPOSTE

Forse la mia domanda può sembrare un po' ingenua, ma la pongo lo stesso.

Per quale motivo l'utente di un computer Commodore "deve" acquistare la vostra rivista? E quali sono i motivi per cui l'autore di un programma deve sentirsi in dovere di inviare il listato a voi e non ad altre riviste del settore?

(Riccardo D'Angelo - Verona)

Precisiamo anzitutto che nessuno "deve" sentirsi in dovere di far qualcosa. Noi ci rivolgiamo agli utenti di sistemi Commodore per evitare che le loro meravigliose macchine corrano il rischio di venir abbandonate in un cassetto dopo le prime inevitabili difficoltà. Siamo convinti che nessuno è un genio e che chiunque, a patto di esser ben guidato, può raggiungere una competenza per nulla inferiore a quella dei "professionisti".

Accontentiamo comunque il lettore rispondendo alla domanda per nulla banale.

1/ Il nome Commodore nella nostra testata indica chiaramente che vengono trattati esclusivamente argomenti che riguardano tali computer. Notizie su altre macchine, di interesse pressocchè nullo per i possessori di un Vic o di un 64, non partecipano ad aumentare il numero delle pagine e, di conseguenza, il prezzo della rivista.

2/ Le 2.500 lire occorrenti per l'acquisto del fascicolo rappresentano, considerando l'esclusività degli argomenti trattati, il prezzo più basso per entrare in possesso, *mensilmente,* di notizie, programmi, articoli di didattica.

3/ Al contrario di altre riviste del settore, di ogni articolo e programma pubblicati viene *sempre* indicato il nome, cognome, indirizzo dell'autore. Tale decisione è stata presa affinchè gli autori, e solo loro, possano raccogliere i frutti del proprio lavoro. E' infatti frequente il caso di lettori che, per vari motivi, non vogliono trascrivere i programmi dalla rivista e si rivolgono direttamente agli autori dei listati per l'acquisto dei programmi su supporto magnetico.

Un altro motivo per cui vengono pubblicati i nomi degli autori è quello di indicare, anche se indirettamente, alle ditte di informatica che li ricercano, nominativi validi cui affidare incarichi di vario genere: stesura di programmi, dimostrazioni presso rivendite, installazioni di sistemi presso neo-acquirenti eccetera. Alcuni tra i lettori, da noi giudicati elementi molto validi, vengono addirittura indicati ad operatori del settore che chiedono continuamente alla Redazione nominativi di persone "in gamba". Un paio di lettori che hanno voluto seguire i nostri consigli, percepiscono ormai dalle 80 alle 150 mila lire al giorno...

4/ I programmi che compaiono sulla rivista sono *garantiti* funzionanti: nessun listato viene pubblicato se non precedentemente testato.

5/ Commodore Computer Club è l'unica rivista che risponde alle domande dei lettori per telefono: un mezzo rapido ed efficace per ottenere informazioni in... tempo reale.

6/ Gli articoli trattati sono alla portata di *tutti:* anche di colui che ha comprato ieri il suo Vic 20.

7/ I lettori più smaliziati invece, troveranno sempre articoli e programmi che li possano interessare: la rivista, nel selezionare i listati, pensa anche a loro.

8/ Varie iniziative (alcune delle quali ancora... segrete), vengono intraprese per rendere ancora più interessante la rivista.

9/ La Redazione promuove *gratuitamente* la diffusione di notizie riguardanti Club di vecchia e nuova formazione: basta telefonare per approfittare di tale opportunità.

10/ Per ora basta così: gli altri motivi validi per acquistare la nostra rivista li trovi sfogliandola...

Alessandro De Simone

COMMODORE 64, ORA CHE CE L'HAI...

GUARDA CHE CI FAI

MAGIC DESK: SCRIVE, ARCHIVIA, STAMPA O CESTINA.

Solo Commodore ti dà Magic Desk. Guarda che facile. Sullo schermo appare una mano in animazione. Prendi il joy-stick e la muovi indicando gli oggetti della scrivania. Con sole due dita dai tutte le istruzioni. Macchina per scrivere: il tuo Commodore 64 adesso funziona come una vera macchina per scrivere elettrica, con tanto di effetti sonori. Cassetti: archivi le pagine che hai battuto in uno dei classificatori. E le richiami quando vuoi.

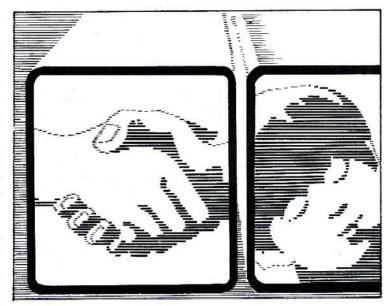
Stampante: le tue pagine sono stampate automaticamente sulla stampante Commodore.

Cestino: il computer cancella tutto quello che non ti serve. Orologio. Hai il costante controllo del tempo mentre lavori con Magic Desk. Non è semplicemente favoloso?

UN PO' DI ORDINE FRA I BIT

L'articolo pubblicato nel numero scorso (Tecniche di overlay) descriveva, in maniera succinta, il modo in cui il Vic ed il Commodore 64 gestiscono i dati, ed in particolare le variabili.

In queste righe illustreremo in modo più approfondito i concetti appena accennati servendoci di una serie di programmini semplici e di immediata comprensione utilizzabili sul Vic 20 come sul Commodore 64.

Esame dei puntatori (programma 1): dovrebbe esser noto che i byte 43 e 44 puntano all'inizio del programma basic, mentre 45 e 46 puntano alla sua fine. In effetti i byte 45 e 46 puntano all'inizio delle variabili, ma, dato che queste sono allocate a partire dal byte successivo all'ultima locazione basic, offrono comunque una valida indicazione per individuare il termine di un programma. Digitate il programma 1 e battete RUN. Esaminiamo ciò che accade (figura 1) non senza aver dato dapprima una definizione: diremo che una variabile è "dichiarata" quando viene nominata per la prima volta

nel corso di un'elaborazione all'interno di un programma basic. Ciò significa che nel caso del microprogramma che segue:

```
100 A=100: B=3.456:
A$="PROVA"
```

La prima variabile ad essere dichiarata è "A", la seconda "B" e la terza A$. Osserviamo, invece, il caso seguente:

```
100 A=100
110 GOSUB 200
120 B=456: C$="PRIMO"
130 . . .
140 . . .
200 C=34.87: D$="SECONDO"
210 RETURN
```

Apparentemente l'ordine di dichiarazione delle variabili sembra essere:

A, B, C$, C, D$

Seguendo invece la struttura *logica* del programma, il computer

COMPUTER QUESTO MESE È QUESTO

COMPUTER

N.65 - lire 3000 il "newsmagazine" dell'informatica

VisiCalc cos'è, come lavora

●

Ditelo con i grafici

●

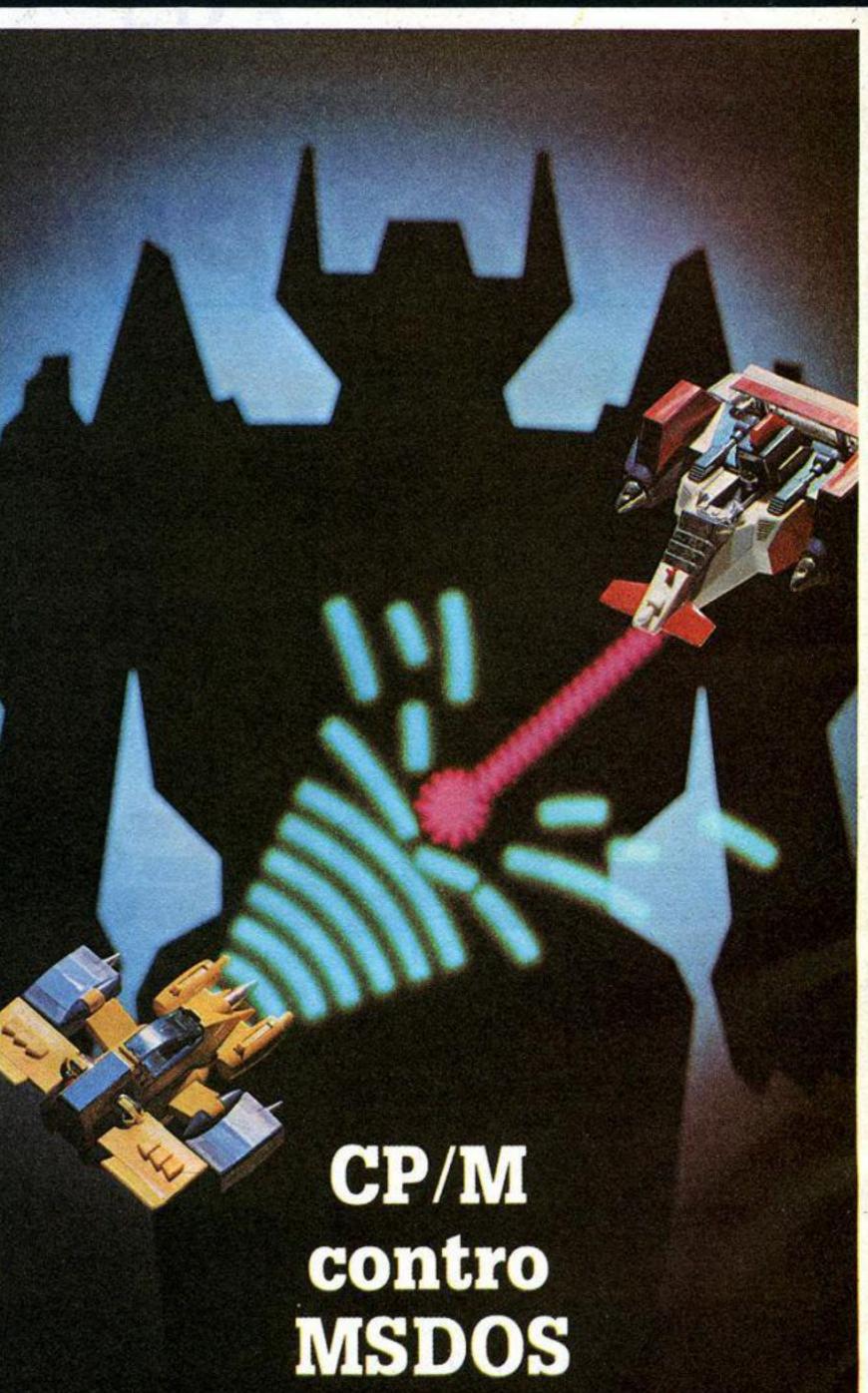
Hardware? No, Superware

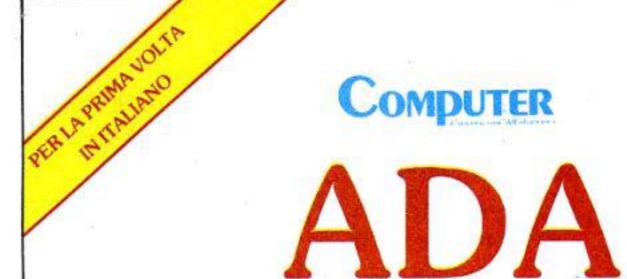
●

"C", linguaggio per gli anni '80

Macintosh by Apple

CP/M contro MSDOS

Computer - Anno VIII n° 65 - 25 marzo 1983 - Sped. Abb. Post. Gr. III/70 Distr. MePe

Computer questo mese é questo...

E QUESTO ....

Il linguaggio passe-partout dei computer degli anni '80

incontra dapprima la variabile A, ma, subito dopo, grazie a GOSUB 200, memorizza "C" e "D$". Solo al "ritorno" dalla subroutine incontra "B" e "C$".

Quanto detto giustifica il motivo per cui, nei listati pubblicati, si è preferito comunicare immediatamente, nelle primissime righe, i nomi delle variabili che saranno adoperate nel corso dell'elaborazione dei singoli programmi. Il sistema operativo, come infatti vedremo, alloca una in coda all'altra le variabili che a mano a mano incontra.

Torniamo al programma 1 ed esaminiamolo nei dettagli:

Riga 130 - dichiarazioni variabili. Si noti la variabile A$ realizzata servendosi di una concatenazione. Ciò serve per "costringere" il sistema operativo ad allocare i byte costituenti la stringa stessa in una zona estranea all'area del programma Basic.

Righe 140 - 160. Dimensionamento di tre vettori di variabili intere, virgola mobile e stringhe lunghi ciascuno 20 elementi (non si dimentichi infatti la posizione "zero").

Righe 170 - 305. Stampa dei risultati, la visualizzazione (riportata nella figura 1) evidenzia in modo chiaro lo spazio occupato dalle variabili all'interno della memoria RAM.

Si precisa che i risultati di figura 1 si riferiscono al Commodore 64. I possessori di Vic 20 noteranno altri valori. Ciò è dovuto al fatto che gli indirizzi di partenza dei due computer sono diversi. Gli stessi utenti del CBM 64 potranno notare valori differenti. Il motivo deve essere ricercato nel fatto che i valori indicati cambiano a seconda della lunghezza del programma basic. E' infatti sufficiente che, nella trascrizione del programma, un solo byte venga digitato in più o in meno affinchè la lunghezza cambi (caso dei REM non trascritti, spazi bianchi tra istruzioni, eccetera).

Proprio a tal proposito il lettore può verificare le differenze esistenti digitando un maggior numero di righe, variabili, matrici, stringhe. Ad ogni modifica apportata si potranno notare varie cose interessanti. Ne accenniamo alcune:

1/ aumentando o diminuendo il numero di righe basic, o alterando la loro lunghezza, l'inizio del basic non varia mai.

2/ Apportando variazioni viene modificato, invece, l'indirizzo dell'ultimo byte basic.

3/ Se viene variato il puntatore di fine basic, vengono modificati *tutti* gli altri puntatori della quantità eguale alla variazione della lunghezza del programma.

4/ Ogni variabile (intera, decimale, stringa) occupa *sempre* sette byte. Si deduce che la differenza tra i valori dei puntatori di fine ed inizio variabili è sempre un multiplo di sette (oppure vale zero nel caso non siano state dichiarate variabili).

5/ I puntatori di inizio e fine stringhe coincidono nel caso in cui le stringhe dichiarate siano allocate all'interno dell'area del programma basic. Se invece, come nel listato presentato, esse rappresentano il risultato di una elaborazione di stringhe (LEFT$, RIGHT$, somme, eccetera), la differenza tra i puntatori coinciderà col numero dei caratteri costituenti le stringhe dichiarate.

6/ I vettori occupano uno spazio diverso a seconda della propria tipologia.

• *Vettori di variabili intere.* Ogni vettore occupa il numero di byte seguenti: due per il nome del vettore o matrice pluridimensionale; due per indicare il numero di byte occupati dall'intero vettore; una per indicare il numero delle dimensioni (max = 256); due per ciascuna dimensione: ognuna di tali coppie ha il compito di indicare il numero di valori occupati dalla dimensione interessata.

Per ogni vettore dichiarato di variabili intere si ha pertanto un minimo di sette byte (con funzioni di "indice") necessari, al sistema operativo, per ottenere informazioni sul vettore stesso. Oltre a quelli esaminati bisogna, ovviamente, aggiungere due byte per la memorizzazione di ciascun valore del vettore. Come è noto con due byte è possibile memorizzare valori interi compresi tra -32768 e +32767.

• *Vettori o matrici di valori decimali.* Il numero di byte di "indice" sono gli stessi di quelli delle matrici intere (minimo 7). Cambia il numero dei byte per ciascun valore decimale: cinque invece di due.

• *Vettori stringhe.* Idem per i

...e a maggio

TI REGALA QUESTO

BUONO FLOPPY

BUONO FLOPPY
BUONO FLOPPY

Corri dal più vicino distributore Rhône-Poulenc Systemes (vedi l'elenco all'interno della rivista) e riceverai un dischetto, Flexette® da 5" 1/4. In alternativa potrai ottenere uno sconto del 15% su una confezione di 10 dischetti da 5" 1/4 o da 8".

Perchè sia valido il presente coupon va completato in tutte le sue parti.

COMPUTER

quando smanettare non basta

AREA BASIC
N. DI BYTE
FUNZIONE
2
Link linea Basic successiva
2
Numerazione linea
0÷80
Istruzioni Basic
ULTIMO
Codice termine linea (0)
ULTIMI TRE
Codice termine programma (0,0,0)
BASIC
AREA VARIABILI
(7 BYTE PER CIASCUNA)
N° BYTE
FUNZIONE
2
NOME VARIABILE
VALORE INTERO
3-4
VALORE INTERO
VALORE DECIMALE
3-7
VALORE DECIMALE
STRINGA
NUMERO CARATTERI
INDIRIZZO PRIMO CARATTERE
TIPO DI VARIABILE
PRIMO BYTE
INTERA
CODICE ASCII
DECIMALE
CODICE ASCII
STRINGA
CODICE ASCII
NOME
100 AB% = 100:AB = 10.3:AB$ = "PROVA"
110 DIM AB (1,1)
120 DIM AB$ (1,1)
PAGINA ZERO
ZONA DELLE MEMORIE VOLAT
BASIC
VARIABILI
MATRICI-VETTORI
TAVOLA DEGLI INDIRIZZI
PEEK (X)+PEEK (X+1)*256
X
X+1
A 43 44 Inizio Basic
B 45 46 Inizio variabili
C 47 48 Inizio matrici
D 49 50 Fine matrici
E 51 52 Inizio stringhe
F 53 54 Fine stringhe
G 55 56 Fine mem. RAM

byte di indice; il numero per ciascun elemento del vettore è fissato invece in tre: il primo indica il numero dei caratteri della stringa esaminata, gli altri due individuano l'indirizzo del primo byte in cui è allocato il primo di essi.

Il lettore, per verificare quanto asserito, può modificare a piacimento le linee 130-160 inserendo o cancellando linee, dichiarando altre variabili, dimensionando in modo vario più matrici. Dopo aver effettuato la variazione, digitando RUN, sarà facile effettuare un controllo sui mutamenti avvenuti, specialmente per ciò che riguarda gli indirizzi di inizio e fine basic, variabili, matrici e stringhe.

Nel caso particolare del programma N.1 le tre variabili dichiarate (A, A$, A%) rendono appunto pari a 3*7=21 la zona RAM dedicata ad esse. Il vettore A $ (19), d'altra parte, occupa 7+3*20+=67 byte, mentre A(19) richiede 7+20*5=107 e A%(19) 7+20*2=47 locazioni di memoria per un totale di 221 byte. Tale valore, aggiunto all'indirizzo di inizio matrici, fornisce appunto il valore del puntatore di fine matrici.

Prima fase

Esame variazione puntatori (programma 2).

Il secondo listato presentato ha lo scopo di dimostrare che i puntatori del sistema operativo vengono alterati *durante* l'elaborazione dello stesso programma. Si noti infatti la subroutine 300-350: questa, quando viene richiamata, visualizza su schermo i valori dei puntatori che sono attivi in quel particolare momento dell'elaborazione. Si noti inoltre che per il calcolo si è evitato il ricorso a variabili, proprio per rendere più comprensibile il listato stesso. Esaminiamolo in dettaglio:

Riga 140. Prima che una qualsiasi variabile venga dichiarata, si utilizza la subroutine 300. Il risultato dimostra che, almeno in questa prima fase, i puntatori di inizio e fine variabili coincidono, proprio perchè non ve ne sono.

Riga 150. Vengono dichiarate due variabili intere (AA% BB%).

Righe 160-170. Il nuovo rinvio alla subroutine 300 evidenzia l'occupazione di 14 byte nell'area destinata alle variabili.

Righe 170-180. Nuovo incremento dei puntatori di fine variabili (6 dichiarate fino a questo momento).

Righe 190-200. Dichiarazione dei valori di due variabili precedentemente dichiarate in riga 170. Si noti che i puntatori rimangono inalterati dimostrando che, una volta che una variabile numerica viene dichiarata, successive modifiche del suo contenuto *non* alterano il numero di byte destinati all'occupazione da parte delle variabili interessate.

Righe 220-280. Utilizzando le stesse variabili che individuano i puntatori e ricorrendo ad un ciclo FOR...NEXT con STEP di 7 (multiplo di occupazione dei byte), vengono visualizzati i contenuti delle locazioni di memoria riservate alle variabili.

La prima delle due lettere che compaiono (in reverse) rappresentano il nome della variabile. Tra parentesi sono raffigurati i loro valori del codice interno Commodore. L'ultimo dato (in reverse) del primo rigo rappresenta l'indirizzo del primo byte contenente il nome della variabile. Al rigo successivo vengono visualizzati i contenuti dei sette byte interessati dalla variabile. Fate attenzione al modo di allocare il nome delle variabili. Si noti inoltre il fatto che, nel caso di variabili intere o stringa, alcuni byte (gli ultimi) sono *sempre* posti al valore nullo.

Anche in questo caso il lettore, data la versatilità del programma proposto, può divertirsi a modificare la riga 150 inserendo altre variabili intere, decimali, stringa dai nomi più bizzarri: il segmento di programma 220-270 visualizzerà in ciascun caso ciò che succede all'interno del calcolatore quando vengono dichiarate le tre tipologie di variabili.

Proviamo ora, avendo in memoria ancora il programma 2, a dichiarare (riga 150) le seguenti tre variabili dallo stesso nome:

AA%, AA, AA$

Una volta dato il consueto RUN ci accorgiamo che nel primo caso il nome viene visualizzato come due lettere A maiuscole, nel secondo con due minuscole mentre nel terzo caso la prima è maiuscola e la seconda minu-

```
100 REM ***        STUDIO DEI PUNTATORI:          ***
110 REM *** QUANTITA' DI MEMORIA OCCUPATA DAL ***
120 REM ***  BASIC  E  DA DIVERSE VARIABILI.   ***
130 A=12:A$= "" + "1234567890":A%=123
140 DIMA$(19):FORA=0TO19:A$(A)=A$:NEXT
150 DIMA(19):FORA=0TO19:A (A)=A :NEXT
160 DIMA%(19):FORA=0TO19:A%(A)=A :NEXT
170 PRINT"INIZIO BASIC    " PEEK(43)+PEEK(44)*256
180 PRINT"FINE   BASIC    " PEEK(45)+PEEK(46)*256
190 PRINT"INIZIO VARIABILI"
200 PRINT"FINE DELLE VAR." PEEK(47)+PEEK(48)*256
210 PRINT"INIZIO MATRICI"
220 PRINT"FINE MATRICI    " PEEK(49)+PEEK(50)*256
230 PRINT"INIZIO STRINGHE" PEEK(51)+PEEK(52)*256
240 PRINT"FINE STRINGHE   " PEEK(53)+PEEK(54)*256
250 PRINT"FINE MEMORIA    " PEEK(55)+PEEK(56)*256
260 PRINT"1 VARIABILE INTERA, 1 IN VIRG.MOB"
270 PRINT"1 VARIABILE STRINGA DI 10 CARATTERI"
280 PRINT"1 VETTORE DI 20 VALORI INTERI ";
285 PRINT"(2*20+7)":REM A. DE SIMONE
290 PRINT"1 VETTORE DI 20 VALORI DECIMALI";
295 PRINT" (5*20+7)"
300 PRINT"1 VETTORE DI 20 STRIN.(10 CAR.)";
305 PRINT" (3*20+7)"

READY.
```

```
INIZIO BASIC     2049

FINE   BASIC     2950
INIZIO VARIABILI

FINE DELLE VAR.  2971
INIZIO MATRICI

FINE MATRICI     3192

INIZIO STRINGHE  40750

FINE STRINGHE    40760

FINE MEMORIA     40960

1 VARIABILE INTERA, 1 IN VIRG.MOB
1 VARIABILE STRINGA DI 10 CARATTERI
1 VETTORE DI 20 VALORI INTERI (2*20+7)
1 VETTORE DI 20 VALORI DECIMALI (5*20+7)
1 VETTORE DI 20 STRIN.(10 CAR.) (3*20+7)
```

```
100 REM ***     ALLOCAZIONE DELLE VARIABILI NUMERICHE         ***
110 REM *** PRIMA FASE: ESAME PUNTATORI DEL BASIC E DELLE ***
120 REM ***      VARIABILI NEL CASO DI 6 DICHIARAZIONI       ***
130 :
140 PRINT CHR$(14) "PRIMA DI DICHIARAZIONI": GOSUB 300
150 AA%=100: BB%=32767
160 PRINT "DOPO DUE DICH."
170 GOSUB 300: PI=PI: PF=PF: I=I: J=J
180 PRINT "DOPO 6 DIC.": GOSUB 300
190 PI=PEEK(45)+PEEK(46)*256
200 PF=PEEK(47)+PEEK(48)*256: GOSUB 300
205 :
210 PRINT "  ELENCO VARIABILI     "
215 :
220 FOR I=PI TO PF-7 STEP 7:PRINT "";
230 PRINT CHR$(PEEK (I)) CHR$(PEEK(I+1))" (";
240 PRINT PEEK(I) PEEK(I+1)")";
250 PRINT " "I
260 FOR J=I TO I+6: PRINT PEEK(J);:NEXT J
270 PRINT: NEXT I
280 PRINT: PRINT AA%;BB%;PI;PF;I;J
290 END:REM A. DE SIMONE / DIDATTICA '84
295 REM SUBROUTINE DI ESAME PUNTATORI
300 PRINT "INIZIO DEL BASIC" PEEK(43)+PEEK(44)*256
310 PRINT "FINE BA.=IN.VAR." PEEK(45)+PEEK(46)*256
320 PRINT "FINE VARIABILI  " PEEK(47)+PEEK(48)*256
330 PRINT "PREMI UN TASTO"
340 IF PEEK(197)=64 THEN 340
350 RETURN
360 :
370 :
380 :            PROGRAMMA N.2

READY.
```

```
100 REM *** SECONDA FASE: MODIFICA BYTE ***
110 REM *** RELATIVI A VARIABILI INTERE ***
120 :
130 AA%=100: PRINT CHR$ (14)
140 PRINT"PUNTATORI VARIABILE": GOSUB 360
150 PRINT"1) MODIFICA AA."
160 PRINT"2)MODIFICA PUNTATORI"
170 GET A$: IF A$="1" THEN GOSUB 220:GOTO 150
180 IF A$="2" THEN GOSUB 260: GOTO150
190 GOTO170
200 :
210 REM *** RINE MODIFICA VALORE DI AA%
220 INPUT"AA%=";AA$: IF AA$="←" THEN RETURN
230 AA=VAL(AA$): IF AA>32767 OR AA<-32768THEN220
240 AA%=AA: GOSUB 360: GOTO 220
```

```
250 REM *** RINE MODIFICA BYTE VAR. AA%
260 INPUT "BYTE";AA$:AA=VAL(AA$)
270 IF AA$="←" THEN RETURN
280 IF AA >I+6 OR AA <I THEN260
290 A1=AA: REM A. DE SIMONE DIDATTICA '84
300 INPUT "VALORE";AA$:AA=VAL(AA$)
310 IF AA$="←" THEN 260
320 IF AA >255 OR AA <0 THEN300
330 POKE A1,AA: GOSUB 360: GOTO 260
340 :
350 REM *** VISUALIZZAZIONE BYTE VAR.INTERA AA%
360 I=PEEK(45)+PEEK(46)*256: X1=PEEK(I): X2=PEEK(I+1)
370 PRINT "■" I;X1 CHR$ (X1)
380 PRINT "■" I+1 X2 CHR$ (X2)
390 FOR J=I+2 TO I+6: PRINT "■" J "■" PEEK(J): NEXT
400 PRINT"■■AA%■="AA% "■■■■BB%■="BB%
410 RETURN
```

```
100 REM *** TERZA FASE : MODIFICA BYTE RELATIVI ***
110 REM ***      A VARIABILI IN VIRGOLA MOBILE    ***
120 :
130 AA=100: PRINT CHR$ (14)
140 PRINT"■PUNTATORI VARIABILE": GOSUB 360
150 PRINT"■1) MODIFICA AA"
160 PRINT"2)MODIFICA PUNTATORI■"
170 GET A$: IF A$="1" THEN GOSUB 220:GOTO 150
180 IF A$="2" THEN GOSUB 260: GOTO150
190 GOTO170
200 :
210 REM *** RINE MODIFICA VALORE DI AA
220 INPUT"■AA=";AA$: IF AA$="←" THEN RETURN
230 AA=VAL(AA$)
240 GOSUB 360: GOTO 220
250 REM *** RINE MODIFICA BYTE VAR. AA
260 INPUT "■BYTE";AA$:AA=VAL(AA$)
270 IF AA$="←" THEN RETURN
280 IF AA >I+6 OR AA <I THEN260
290 A1=AA: REM A. DE SIMONE DIDATTICA '84
300 INPUT "VALORE";AA$:AA=VAL(AA$)
310 IF AA$="←" THEN 260
320 IF AA >255 OR AA <0 THEN300
330 POKE A1,AA: GOSUB 360: GOTO 260
340 :
350 REM *** VISUAL.BYTE VAR.VIRGOLA MOBILE AA
360 I=PEEK(45)+PEEK(46)*256: X1=PEEK(I): X2=PEEK(I+1)
370 PRINT "■" I;X1 CHR$ (X1)
380 PRINT "■" I+1 X2 CHR$ (X2)
390 FOR J=I+2 TO I+6: PRINT "■" J "■" PEEK(J): NEXT
400 PRINT"■■AA■="AA "■■■■BB■="BB
410 RETURN

READY.
```

COMMODORE 64

HOTLINE · UPDATE · GARANZIA

Tre nuove parole nel campo dell'informatica. Esse rappresentano il

- NUOVO SERVIZIO -

che la Leoni Informatica, prima fra tutti offre ai suoi clienti. COMMODORE 64

HOTLINE — linea telefonica dedicata alla risoluzione dei problemi dei clienti. Chiamando il numero telefonico riservato che troverete sulla cartolina garanzia acclusa ai programmi, riceverete tutte le informazioni che vi necessitano.

UPDATE — servizio di aggiornamento continuo dei programmi acquistati. Ogni modifica ai programmi realizzati dalla Leoni Informatica sarà fornita agli utenti degli stessi.

GARANZIA — tutti i programmi Leoni Informatica sono coperti da garanzia a Vita contro guasti di origine.

ALCUNI PROGRAMMI PER COMMODORE 64

Cod.	Descrizione	Prezzo
0047	Gestione Anagrafiche	120.000
0050/c	Totocalcio Sviluppo Colonnare	80.000
0051/c	Gestione dei conti Casa	100.000
0055/c	Impariamo il basic	100.000
0056/c	Dichiarazione Iva	60.000
0063/c	Cento Programmi per il 64	80.000
0064	Compilatore Pet Speed	80.000
0065/c	Fido Clienti	100.000
0066	Conto Corrente	150.000
0067	Gestione piano dei Conti	150.000
0068	Gestione Appuntamenti	150.000
0071	Gestione Ordini	150.000
0086	Gestione Librerie	150.000
0090	Mailing List	150.000
0091	Rubrica Telefonica	120.000
0094/c	Gestione Scheda 4800 car.	160.000
0096	Gestione Scheda Agganciata al Mailing List	250.000
0116	Scadenziario Effetti	200.000
0120	Contabilità Fatture Iva/Imponibile	200.000
0121	Contabilità Semplice	400.000
0136	Legge 373	150.000
0143	Magazzino Grossisti (2500 art.)	380.000
0144	Magazzino Fatturazione Agganciate	400.000
0148	Gestione Ottici	300.000
0149	Gestione Dentisti	300.000
0151	Gestione Farmacie	400.000
0152	Gestione Studi Medici	300.000
0158	Magazzino Dettaglio (2500 art.)	380.000
0159	Magazzino Taglia e Colore	380.000
0160	Bolle e Fatture	300.000
0162/c	Screen Grafix	150.000
0163	Copia Disco Singolo	50.000
0165/c	Assembler Disassembler	80.000
0169/c	Magazzeno alfanumerico (1100 articoli)	250.000
0080	Gestione Clubs Nautici	250.000
0081	Gestione Officine	400.000
0087	Gestione Ristoranti	400.000
0131	Gestione Hotel/Pensioni	400.000
0132	Gestione Parrucchieri	400.000
0133	Gestione Gommisti	400.000
0170	Gestione Tavola Calda	400.000
0171	Gestione Lavanderia	400.000
0155	Gestione Condominio	300.000
0166/c	Compactor	50.000
0167/c	Scompactor	50.000
0168/c	PET Emulator	35.000
0306/c	Character editor	40.000
0309/c	Hires image	40.000
0310/c	Hard copy	40.000
0312	Master 64	225.000
0313/c	Tool 64	85.000
0314/c	Stat 64	65.000
0157	Calc Result Advanced	350.000
0319	Easy Script	125.000
0322/c	Forth 64	65.000

I programmi Leoni sono disponibili presso tutti i punti vendita MELCHIONI ELETTRONICA
Vendita per corrispondenza anche dell'HARDWARE

S.r.l.
20142 MILANO - VIA DON RODRIGO, 6 - TEL. 02/8467378

```
100 REM *** QUARTA FASE: ESAME ALLOCAZIONE ***
110 REM *** MATRICI VARIABILI INTERE E NON ***
120 :
130 X1=X1:X2=X2:X3=X3:X4=X4:X5=X5:I=I
140 :
150 DIM AA%(12,24),CA(12),CX%(32)
160 :
170 PRINT CHR$(14)
180 X1=PEEK(47) + PEEK(48)*256: X2=PEEK(49)+PEEK(50)*256
190 PRINT "■ULTIMO BYTE OCCUPATO■" X2
200 PRINT"■"X1 PEEK(X1) CHR$(PEEK(X1)) "■ NOME"
210 PRINT "■" X1+1 PEEK(X1+1) CHR$(PEEK(X1+1)) "■ VETTORE"
220 PRINT "■" X1+2 PEEK(X1+2) "■ ...BYTE"
230 PRINT "■" X1+3 PEEK(X1+3) "■ ...OCCUPATI=";
240 X4=PEEK(X1+2)+PEEK(X1+3)*256:PRINTX4
250 X5=X1+PEEK(X1+2)+PEEK(X1+3)*256
260 PRINT "(OCCUPAZIONE FINO A" X5-1 ")"
270 PRINT "■" X1+4 "■" PEEK(X1+4)" DIMENSIONI=";
280 X3=X1+4: PRINT "■" PEEK(X3)
290 FOR I=1 TO PEEK(X3)*2: PRINT"■"X3+I+1;PEEK(X3+I):NEXT
300 IF X2-X1 = X4 THEN 350
310 X1=X5: REM A. DE SIMONE DIDATTICA '84
320 PRINT "■PREMI UN TASTO"
330 IF PEEK(197)=64 THEN 330
340 GOTO 180
350 LIST 150

READY.
```

```
100 REM *** STUDIO ALLOCAZIONE STRINGHE ***
110 :
120 AA$ = "STRINGA"
130 X1=X1: X2=X2: I=I: PRINT "■"
140 X1 = PEEK(45)+PEEK(46)*256
150 X2 = PEEK(47)+PEEK(48)*256
160 PRINT X1 "■" CHR$(PEEK(X1)) "■ NOME"
170 PRINT X1+1 "■"CHR$(PEEK(X1+1)) "■ STRINGA"
180 PRINT X1+2 "LUNG. STRINGA =■" PEEK(X1+2)
190 PRINT X1+3 "LA.INDIR.STRINGA=■" PEEK(X1+3)
200 PRINT X1+4 "HA.INDIR.STRINGA=■" PEEK(X1+4)
210 PRINT"■■ " PEEK(X1+3) "■ +" PEEK(X1+4)"■ * 256 =";
220 X2=PEEK(X1+3)+PEEK(X1+4)*256:PRINTX2"■)■"
230 FOR I=0 TO PEEK(X1+2)-1:REM A.DE SIMONE
240 PRINT "■"X2+I"■" CHR$(PEEK(X2+I)): NEXT
250 LIST 120

READY.
```

scola. Questo modo di alterare la prima, la seconda o entrambe le lettere delle variabili, consente al sistema operativo del computer di capire quale dei tre tipi di variabili ha individuato.

Seconda fase

Alterazione del contenuto dei byte interessati all'immagazzinamento delle variabili.

Finora abbiamo assistito "passivamente" al modo in cui il calcolatore gestisce la memoria RAM nel definire variabili di qualunque tipo. Vediamo ora che succede se proviamo intenzionalmente, ad alterare il loro contenuto. A tale scopo digitiamo il programma N. 3. In esso (riga 130) viene definita una sola variabile, ma sarà più che sufficiente per lo scopo proposto. E' ovvio che il lettore, con la propria fantasia, può apportare tutte le sofisticazioni che ritiene opportune.
Righe 140-190. Menu di scelta. Viene chiesta una delle due scelte possibili:
1/ alterazione del valore di AA% (dichiarata in riga 130);
2/ alterazione dei singoli byte costituenti la variabile.

Nel caso si scelga l'opzione 1, verrà richiesto un nuovo valore da attribuire alla variabile AA% (righe 220-240). E' ovvio che vengono rifiutati valori che escono dall'intervallo -32768 +32767 (riga 230). Subito dopo il valore digitato viene depositato in AA% e la subroutine 360 viene incaricata di visualizzare non solo il nome della prima variabile dichiarata (che, guarda caso, è proprio AA%), ma anche il contenuto dei singoli byte relativi ad essa oltre al loro indirizzo.

Digitando il carattere di freccia a sinistra (←) si ritorna al menu principale. Se ora si sceglie l'opzione 2) si avrà la possibilità di alterare il contenuto dei singoli byte. Modificandoli a caso si possono fare le seguenti deduzioni:
a/ alterando il contenuto degli ultimi tre byte il valore di AA% non viene in alcun modo modificato. Ciò dimostra che il sistema operativo, nel caso di variabili intere, "guarda" esclusivamente il terzo ed il quarto byte.
b/ Alterando il contenuto di questi due byte si ottengono (riga 400) valori diversi di AA%.
c/ Alterando i primi due byte (relativi cioè al nome della variabile) ci accorgiamo che il computer non riconosce più la prima variabile che incontra col nome AA%, ma col nome che le abbiamo attribuito! Provate dunque (avendo davanti a voi la tabella del codice ASCII e addizionando 128 al valore del carattere tabellato) a modificare in tal senso i due byte, dapprima trasformando AA% in BB% (cioè 193-193 in 194-194) e poi in altre lettere qualunque.

Terza fase

Modifica dei byte relativi a variabili decimali.

Il programma 4 è sostanzialmente identico a quello N. 3. Le differenze consistono nel trattamento di valori numerici diversi.

Quarta fase

Esame allocazione delle matrici.

Il listato 5, che non viene descritto in maniera approfondita, ha lo scopo di dimostrare che l'allocazione dei vettori segue le regole prima illustrate. Il lettore può verificarlo alterando la riga 150 inserendo più vettori o dimensionandoli diversamente. L'automatismo del programma consente di esaminare, byte dopo byte, ciascun valore dichiarato. Diremo soltanto che, anche in questo caso, i primi vettori che si incontrano sono proprio quelli dichiarati seguendo l'ordine "logico" del programma.

Quinta fase

Studio dell'allocazione delle stringhe.

In quest'ultimo listato (N. 6) viene esaminato il luogo in cui i caratteri delle singole stringhe dichiarate vengono depositati. Il lettore può provare ad alterare il contenuto della variabile AA$ (riga 120), oppure a modificarlo ricorrendo a concatenazioni di più stringhe. Uno studio particolareggiato sull'allocazione dei vettori stringa non viene indicato dato che, a questo punto, il lettore, seguendo la falsa riga dei programmi pubblicati in queste pagine, può farlo da solo. ▪

Alessandro De Simone

ELIHELP

In un tunnel buio, percorso da missili che passano ogni 15 secondi a diverse altezze, un elicottero deve salvare il maggior numero possibile di omini, evitando le macerie sparse nel tunnel stesso, illuminando il terreno e tenendo d'occhio il consumo del carburante.

Il movimento dell'elicottero è ottenuto con i tasti F1 (in alto), F3 (in basso), F5 (avanzamento), F7 (lancio della scaletta). Quest'ultimo comando è necessario per salvare gli omini stessi, l'elicottero infatti non atterra ma si posiziona a quattro caratteri sopra il livello del suolo e sgancia la scaletta di fianco all'omino da salvare.

Ogni movimento richiede un certo consumo di carburante: mezzo litro per sganciare la scaletta, scendere o illuminare, un litro per ogni avanzamento, un litro e mezzo per salire. All'interno del tunnel esistono però delle piazzole di distribuzione del carburante, se si riesce ad atterrarvi si può fare un "pieno" di venti litri.

In alto sullo schermo, appare il radar e in basso una sezione del radar ingrandita con il disegno dell'elicottero. Si noti che la sezione è di 8 caratteri e il radar di 8 punti (Bit), in altezza.

In questo modo si ha una perfetta corrispondenza tra radar e sezione. Il radar segnala la parte "davanti" del veicolo (essendo di due caratteri).

La luce, come già accennato, illumina da una altezza massima di 3 caratteri dal suolo, distribuendosi a raggiera, se si vola bassi quindi si illumina poco. Avanzando si muove solo la posizione nel radar del veicolo e ovviamente il terreno (buio). Il terreno, cioè il percorso, è definito nella riga 180-182:

- A = ometto;
- "diesis" = carburante;
- B,C,D,E,F = elicottero e scaletta.

Le cifre che appaiono sul video rappresentano il numero degli omini presi, i punti, il carburante disponibile.

Il programma esce dall'alta risoluzione, con istruzione **POKE 36869,240** se questo e altri piccoli accorgimenti falliscono è dovuto alla scarsa memoria: mancano le istruzioni nel programma, i nomi alle cifre, l'uscita dall'alta risoluzione, ecc.

Descrizione

110 - sistema i caratteri definendoli normalmente;

120-175 - definisce i caratteri in hi-res;

177-178 - assegnazioni e inizializzazione variabili;

180-182 - creazione del percorso (modificabile);

198 - azzera i byte dei caratteri che formano il radar;

200 - disegna il radar (come posizione);

220-221 - disegna i contorni del radar e della sezione;

224-228 - riempie il radar (il sistema è simile a quello usato da un programma che disegnava grafici in alta risoluzione pubblicato sul numero 3 di CCC, tempo fa); per modificare ciò che il radar segnala è necessario alterare la riga 225 e in particolare la relazione dell'istruzione IF... THEN.

Come si può notare viene esaminato carattere per carattere il percorso che è nella stringa **D$**. Il meccanismo consiste nel far fare l'istruzione dopo il **THEN** se si vuole che appaia sul radar il simbolo della stringa: se si vuole far apparire tutto tranne gli omini (A) si scrive:

IF W$ "diverso" da "A" THEN...

se si vuole far apparire solo gli omini:

IF W$ = "A" THEN...

se si vuole far apparire omini e

carburante:
IF W$ = "A" OR W$ = "diesis" THEN...
e così via.
297 - assegnazioni variabili radar per l'elicottero;
300-651 - ciclo principale, il loop è pieno di variabili "fantasma" che servono solo a segnalare la presenza di un certo fenomeno;
357 - questa riga controlla il tempo per far partire il razzo ogni 15 secondi, ovviamente modificabile (attenti alla memoria);
610-650 - algoritmo del radar:
L1 = codice del carattere;
T1 = altezza elicottero (come Bit);
Z1 = spostamento orizzontale (come Bit);
K1 = locazione di memoria del Bit risultante;
le variabili sono simili al primo "radar";
1502-1570 - subroutine LUCE;
1 701-1760 - subroutine SCALETTA;
1800-1820 - subroutine STAMPA ELICOTTERO;
1860 - il gioco termina: vengono sommati gli omino-punti e il carbutante risparmiato.

Volendo si possono apportare tanti piccoli accorgimenti (ricorrendo alla tecnica dell'overlay). Il primo programma, in questo caso, potrebbe essere costituito dalla definizione dell'alta risoluzione, risparmiando al secondo molta memoria. ▪

Marco Salvato
Via Trento, 11 - Mestre (VE)
Tel. 041/959259


```
110 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
120 FORI=7168TO7231:READA:POKEI,A:NEXT:DATA255,0,0,0,0,0,0,0,56,
    186,146,124,56,5 6,40,108
130 DATA68,68,124,68,68,68,124,68,255,255,255,255,255,255,255,255
140 DATA7,160,95,167,0,0,0,1,255,32,254,242,246,252,73,254
150 DATA32,32,254,242,246,252,73,254,0,64,255,71,0,0,0,1
170 FORI=7424TO7455:READA:POKEI,A:NEXT:DATA0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,255
175 DATA1,2,63,193,63,2,1,0,255,255,126,66,94,94,66,126
177 CLR:POKE36869,255:N=36876:POKE36878,15:PE=8064:POKE36879,10:
    PRINT"■":CE=387 84:V=150
178 PR=8097:MR=0:B=1:POKE56,28:POKE52,28:J=1:WS=2
180 D$="■ C B A  DE B  C B A     A  DE   A  C B # B   A  C BA# DEB A A
    C   #A "
181 D$=D$+"■      A     # A DE A A #    B AA   DE# B   A    C   A# A B
    C A GF    A   #A"
182 D$=D$+"■      A      B A B # DE AA     B C A BA "
198 FORI=7248TO7423:POKEI,0:NEXT
200 M=10:FORL=0TO22:POKE7702+L,M:M=M+1:NEXT
220 FORI=7966TO7987:POKEI,3:POKEI+198,3:NEXT:TI$="000000"
221 FORI=7680TO7701:POKEI,33:NEXT:FORI=7724TO7745:POKEI,0:NEXT:
    Z=0:L=10:G=11
224 W$=MID$(D$,G,1):IFW$=" "THEN226
225 IFW$<>"A"THENK=7168+L*8+7:POKEK,2↑(7-Z)+PEEK(K)
226 G=G+1:Z=Z+1:IFZ>7THENL=L+1:Z=0:IFL=32THEN297
228 GOTO224
297 Z1=1:T1=4:L1=10
300 P=PEEK(197):IFAO=1ORAX=1THEN320
301 IFP=39ANDPE>8009THENPOKEPE,32:POKEPE+1,32:PE=PE-22:POKEK1,0
    :T1=T1+1:CE=CE-22 :V=V-1.5
310 IFP=47ANDPE<8119THENPOKEPE,32:POKEPE+1,32:PE=PE+22:POKEK1,0
    :T1=T1-1:CE=CE+22 :V=V-.5
320 IFP=32THENX=7:GOSUB1502
324 IFAX=1ORJ1=1THEN330
```

```
325  IFPEEK(PE+23)=35THENAX=1:V=V+20
330  POKEPE,7:POKEPE+1,6:POKEN,180:POKEN,0:POKECE,5:POKECE+1,5
340  IFP=55THENZ1=Z1+1:B=B+1:OP=0:J1=0:V=V-1
350  PRINT"█████████████████████████"MID$(D$,B,22);:IFAO=1THENPOKEPE+89,32
353  PRINT"████"SPC(14)"              ":PRINT"█████"SPC(14)V"■"
356  IFWS=1THEN363
357  IFMID$(TI$,5,2)="15"ORMID$(TI$,5,2)="30"ORMID$(TI$,5,2)="45"
     ORMID$(TI$,5,2)= "58"THEN360
359  GOTO361
360  WA=(INT(RND(1)*7)*22)+7989:WY=WA+22:WS=1
361  IFWS<>1THEN400
363  POKEWY,32:WY=WY-3:POKEWY,34:IFWY=WA+1THENPOKEWY,32:WS=2
364  IFPEEK(WY-1)=5ORPEEK(WY-1)=7THEN1850
400  IFAO=1THENGOSUB1800
410  IFAX=0ORJ1=1THEN580
420  POKECE+23,2:POKEN,250:JX=JX+1:POKECE+23,2:POKEN,0:IFJX=6THENAX=0:
     JX=0:J1=1
580  POKEPE,4:POKEPE+1,5:POKEN,200:POKEN,0:POKECE,5:POKECE+1,5
584  IFOP=1THEN610
585  IFP=63THENXY=2:GOSUB1701
610  IFZ1=8THENL1=L1+1:POKEK1,2↑(7-Z1):Z1=0
620  IFL1=32THEN1860
630  K1=7168+L1*8+(7-T1):IFV<20THENPOKEN,250
650  POKEK1,2↑(7-Z1):IFV<=0THEN1850
651  GOTO300
1502  V=V-1:IFPE>8141THEN1540
1505  POKECE+22,X:POKECE+23,X:POKECE+24,X
1508  IFPE>8119THEN1540
1511  POKECE+43,X:FORI=CE+44TOCE+46:POKEI,X:NEXT:POKECE+47,X
1515  IFPE>8097THEN1540
1520  POKECE+64,X:FORI=CE+65TOCE+69:POKEI,X:NEXT:POKECE+70,X
1540  FORI=255TO128STEP-1:POKEN,I:NEXT:POKEN,0
1545  IFPE>8075THEN1570
1570  RETURN
1701  V=V-.5:FORU=1TO2:IFU=2THENXY=32
1702  IFPEEK(PE+23)=3THEN1740
1705  POKEPE+23,XY:POKEN,190
1708  IFPEEK(PE+45)=3THEN1740
1710  POKEPE+45,XY:POKEN,200
1713  IFPEEK(PE+67)=3THEN1740
1720  POKEPE+67,XY:POKEN,210
1725  IFPEEK(PE+89)=1THENAO=1:OP=1:S=S+20:O=O+1:PRINT"█████"S;O"■"
      :PO=PE+67:GOTO1760
1740  NEXTU
1760  RETURN
1800  POKEPO+22,32:POKEPO,1:PO=PO-22:IFPO=PE+1THENAO=0:POKEPE+23,32
1820  RETURN
1850  PRINT"█████"S:FORI=255TO128STEP-1:POKEN+1,I:NEXT:POKEN+1,0:END
1860  PRINT"█████"S"+"V"="S+V:POKEN,0:END

READY.
```

RESTORE LN

Un'istruzione molto utile che manca al basic del Vic 20 è la RESTORE LN, cioè la possibilità di far eseguire al computer un restore non alla prima linea di data ma ad una qualsiasi linea da noi desiderata.

Dell'utilità di questa funzione ci si accorge dovendo gestire più tabelle di costanti in "frasi" DATA in quanto la semplice istruzione Restore costringe a noiosissimi calcoli per potere rintracciare la tabella successiva. Esistono

```
100 REM *************
110 REM *  TEST  1  *
120 REM *************
130 REM
140 REM SIMULAZIONE
150 REM ERRORE: MANCA
160 REM LA LINEA LN
170 REM
180 RESTORE:PRINT"■"
190 FORK=1TO6
200 READA$:PRINT"■"A$
210 NEXT
220 REM UN ULTERIORE
230 REM READ A$
240 REM CAUSEREBBE
250 REM UN OUT OF
260 REM DATA ERROR
270 LN=315:GOSUB63500
280 READA$:PRINT A$
290 REM
300 END
310 DATAUNO,DUE
320 DATATRE,QUATTRO
330 DATACINQUE,SEI
1000 :
1010 REM INSERIRE QUI
1020 REM DI SEGUITO LA
1030 REM ROUTINE PUBBLICATA
1040 REM (DA RIGA 63500
1050 REM FINO A 63528)

READY.
```

inoltre molti programmi nei quali compare l'istruzione Restore N non presente nel Basic Commodore.

La semplice routine proposta permette di simulare tale istruzione in modo da renderla operativa.

L'idea di principio di questa routine nasce da un esame dei puntatori nella pagina zero del 6502 e da come vengono memorizzate le linee di istruzioni basic nella memoria.

Iniziamo dalle locazioni di me-

moria 43 e 44. Queste indicano l'indirizzo di partenza del testo basic nella memoria ricavabile dalla formula: indirizzo = PEEK (43) + PEEK (44) *256. Tale indirizzo viene calcolato al momento dell'accensione della macchina dal KERNAL nelle locazioni 641 e 642 e poi passato al Basic che pone tale valore 1 nelle locazioni prima citate 43 e 44.

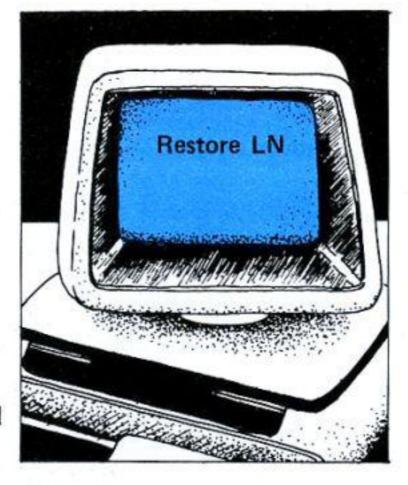
L'indirizzo vale 4097 nel Vic non espanso, ma varia a seconda delle espansioni di memoria RAM inserite.

Passiamo ora ad esaminare come è organizzato il testo basic nella memoria. Prendiamo ad esempio le seguenti istruzioni:

10 REM
20 PRINT
30 END

Se esaminiamo la memoria del Vic potremo osservare la seguente organizzazione:

Possiamo ritrovare tutte le informazioni necessarie per compilare un programma che trovi un numero di linea e che controlli se in quel numero di linea esiste una istruzione DATA, altrimenti ci dia una indicazione di errore.

All'inizio del programma ritroviamo i puntatori 43 e 44 che, guarda caso, nel Vic non espanso puntano alla locazione di memoria 4097 ove si trova l'inizio della prima linea di programma. Possiamo inoltre notare nella tabella che a questo indirizzo troviamo il byte basso del link e in quello successivo il byte alto del link che ci indica dove si trova collocata in memoria la prossima linea basic.

Grazie a questi due byte possiamo istruire la variabile NP che ci permetterà di trovare l'inizio della prossima linea senza dovere scandire tutta la memoria.

La variabile PC indica invece la corrente linea, cioè indica l'indirizzo di partenza della linea che stiamo attualmente esplorando.

Possiamo controllare a questo punto se il numero della linea esplorata è quello che desideriamo e facciamo questa operazione controllando prima il byte alto del numero di linea (in questo modo la routine viene eseguita più velocemente in quanto è meno probabile trovare un byte alto cui corrisponda quello basso errato che non viceversa) e poi quello basso.

Se l'operazione non riesce PC diventa NP cioè passiamo ad esaminare la prossima linea. Se abbiamo trovato il giusto numero di linea controlliamo se c'è effettivamente una istruzione di DATA altrimenti mandiamo un messaggio di errore e terminiamo il programma.

Coca accade se nel programma non esiste la linea specificata in LN? Ad un certo momento il link che manda alla prossima linea di programma in memoria sarà uguale a 0 e con esso la variabile NP. Questo fatto ci avvertirà che il testo basic è finito senza che noi avessimo trovato il numero di linea cercato. Anche in questo caso manderemo un messaggio di errore e termineremo il programma. Se invece tutto sarà andato bene significherà che abbiamo trovato il numero di linea cercato e che quel numero di linea c'era l'istruzione DATA, potremo quindi risalire all'indirizzo in memoria ove esiste l'istruzione data cercata.

E' possibile eseguire questa operazione molto facilmente; prendiamo ad esempio la seguente struttura:

PC AAAA LOW LINK
+ 1 HIGH LINK
Questi puntano a BBBB - NP -
+ 2 LOW N° DI LINEA
+ 3 HIGH N° DI LINEA
+ 4 NNNN DATA
+ 5 A

+ 6 MARCATORE DI FINE LINEA
NP BBBB LOW LINK

E' facile osservare che sarà proprio PC+4 la locazione di memoria che a noi interessa.

Esaminiamo ora i puntatori che ci permetteranno di informare il sistema operativo di dove si troverà la prossima istruzione data da eseguire. Questi si trovano nelle locazioni decimali 63, 64, 65, 66.

Le locazioni 63 e 64 sono rispettivamente il byte basso e alto del numero della linea data, mentre le locazioni 65 e 66 sono il byte basso e alto del puntatore che indica la locazione di memoria dell'istruzione data.

A questo punto il caricamento del puntatore 63 e 64 è immediato in quanto il numero della linea

```
100 REM *************
110 REM *  TEST  2  *
120 REM *************
130 REM
140 REM SIMULAZIONE
150 REM ERR.:NON C'E'
160 REM DATA IN LINEA
170 REM INDICATA
180 RESTORE:PRINT"▘"
190 FORK=1TO6
200 READA$:PRINTA$
210 NEXT
220 REM UN ULTERIORE
230 REM READ A$
240 REM CAUSEREBBE
250 REM UN OUT OF
260 REM DATA ERROR
270 LN=300:GOSUB63500
280 READA$:PRINTA$
290 REM
300 END
310 DATAUNO,DUE
320 DATATRE,QUATTRO
330 DATACINQUE,SEI
1000 :
1010 REM INSERIRE ROUTINE

READY.
```

```
100 REM *************
110 REM *  TEST  3  *
120 REM *************
130 REM
140 REM ESEMPIO DI
150 REM ESECUZIONE
160 REM CORRETTA
170 REM
180 RESTORE:PRINT"▘"
190 FORK=1TO6
200 READA$:PRINTA$
210 NEXT
220 REM UN ULTERIORE
230 REM READ A$
240 REM CAUSEREBBE
250 REM UN OUT OF
260 REM DATA ERROR
270 LN=320:GOSUB63500
280 READA$:PRINTA$
290 READA$:PRINTA$
300 END
310 DATAUNO,DUE
320 DATATRE,QUATTRO
330 DATACINQUE,SEI
1000 :
1010 REM INSERIRE ROUTINE

READY.
```

alla quale c'è il data che interessa lo abbiamo stabilito noi e si trova nella variabile LN, LN(I) contiene il byte alto, mentre la variabile LN(2) contiene il byte basso. Mettere questi due valori nei rispettivi byte del puntatore è un gioco da ragazzi.

Il caricamento del puntatore nei byte 65 e 66 è altrettanto semplice in quanto il programma ha appena calcolato PC + 4 che è appunto il valore con cui va caricato il puntatore in questione.

Una volta inseriti nei puntatori questi valori, quando il basic

incontra una istruzione READ questa va a leggere proprio il primo dato della frase data alla linea da noi specificata, cioè avremo ottenuto proprio l'istruzione RESTORE N che non esiste nel basic Commodore.

Uso della routine

L'uso di questa routine in un programma è alquanto semplice.

Basta infatti caricarla nella macchina prima di digitare il programma che ha bisogno dell'istruzione Restore N, e sostituire l'istruzione RESTORE XXXX con le istruzioni LN = XXXX:

```
100 REM *************
110 REM *  TEST  4  *
120 REM *************
130 REM
140 REM ESEMPIO DI
150 REM UTILIZZO
160 REM
170 REM SCELTA DEL
180 REM NOME
190 RESTORE
200 X=INT(RND(1)*5)+1
210 FORK=1TOX
220 READNO$
230 NEXT
240 REM SCELTA DEL
250 REM VERBO
260 LN=490:GOSUB63500
270 X=INT(RND(1)*3)+1
280 FORK=1TOX
290 READVE$
300 NEXT
310 REM SCELTA DELL'
320 REM OGGETTO
330 LN=520:GOSUB63500
340 X=INT(RND(1)*5)+1
350 FORK=1TOX
360 READOG$
370 NEXT
380 REM STAMPA FRASE
390 PRINT"[illegible]"
400 PRINTNO$:PRINTVE$:PRINTOG$
410 PRINT"[illegible]ANCORA ?"
420 GETA$:IFA$=""THEN420
430 IFA$="S"THENRUN
440 END
450 REM ** NOMI **
460 DATAMARCO,LUISA,E.T.
470 DATAFRANCO,SANDRA
480 REM ** VERBI **
490 DATAROMPE
500 DATAUSA,VUOLE
510 REM ** OGGETTI **
520 DATALA BAMBOLA,UNA CASA
530 DATAUN CUSCINO,IL MARTELLO
540 DATAIL TELEFONO
550 :
1000 REM INSERIRE ROUTINE
READY.
```

GOSUB 63500.

LN deve chiaramente contenere il numero della linea ove eseguire il restore n. Nel programma principale non devono comparire le variabili PC NP TE LN(I) LN(2), PC(I), PC(2) poichè sono usate nella routine.

Consigli utili

Poichè la routine cerca nella memoria l'indirizzo del data interessato sequenzialmente da indirizzi bassi a quelli alti, è evidente che se le frasi data compaiono all'inizio del programma, la routine li troverà più velocemente che non se fossero ubicate alla fine del programma come di solito accade. E' consigliabile perciò una struttura così concepita:

1 GOTO numero di linea ove inizia il programma
2 DATA
3 DATA
X qui inizia il programma

```
63475 REM ********************
63476 REM *                  *
63477 REM * BY MAURO BATTISTI *
63478 REM *                  *
63479 REM * VIA D'ALVIANO 100 *
63480 REM *                  *
63481 REM * 34144 - TRIESTE   *
63482 REM *                  *
63483 REM * TEL. 040/821950   *
63484 REM *                  *
63485 REM ********************
63486 REM
63487 REM
63488 REM IL PROGRAMMA RESTORE LN USA LE
63489 REM VARIABILI LN( 1) LN( 2) TE PC NP
63490 REM PC( 1) PC( 2) CHE PERTANTO NON
63491 REM DEVONO COMPARIRE NEL PROGRAMMA
63492 REM CHE OSPITA QUESTA ROUTINE
63493 REM LA VARIABILE -LN- E' USATA PER
63494 REM COMUNICARE ALLA ROUTINE LA
63495 REM LINEA OVE ESEGUIRE IL RESTORE
63496 REM -- RESTORE LN -- FUNZIONA CON
63497 REM QUALSIASI ESPANSIONE DI RAM
63498 :
63499 :
63500 REM ***********
63502 REM * RESTORE *
63503 REM * LINE    *
63504 REM * NUMBER  *
63505 REM * --LN--  *
63506 REM ***********
63507 :
63508 PC=PEEK( 43)+PEEK( 44)*256
63509 LN( 1)=INT( LN/256)
63510 LN( 2)=LN-LN( 1)*256
63511 NP=PEEK( PC)+PEEK( PC+1)*256
63512 IFNP=0THENGOTO63525
63513 TE=PEEK( PC+3)
63514 IFTE< >LN( 1)THENPC=NP:GOTO63511
63515 TE=PEEK( PC+2)
63516 IFTE< >LN( 2)THENPC=NP:GOTO63511
63517 TE=PEEK( PC+4)
63518 IFTE< >131THENGOTO63527
63519 PC=PC+4
63520 PC( 1)=INT( PC/256)
63521 PC( 2)=PC-PC( 1)*256
63522 POKE63,LN( 2):POKE64,LN( 1)
63523 POKE65,PC( 2):POKE66,PC( 1)
63524 RETURN
63525 PRINT:PRINT"NEL PROGRAMMA NON"
63526 PRINT"ESISTE LA LINEA"LN:END
63527 PRINT:PRINT"ALLA LINEA"LN
63528 PRINT"NON ESISTE UNA"
63529 PRINT"ISTRUZIONE DATA":END

READY.
```

Per risparmiare spazio in memoria si possono eliminare tutti i REM tranne quello nella linea 63500 che va eventualmente sostituito con REM RESTORE LN. ▪

Mauro Battisti
Via d'Alviano, 100
34144 Trieste - Tel. 040/821950

4096	0	Byte sempre a zero	
4097	7	Low Link	- Indicano l'indirizzo 4103 °°
4098	16	High Link	
4099	10	Low numero di linea	
4100	0	High numero di linea	
4101	143	Codice di REM	
4102	0	Marcatore di fine linea	
4103 °°	13	Low link	- Indicano l'indirizzo 4109 = =
4104	16	High link	
4105	20	Low numero di linea	
4106	0	High numero di linea	
4107	153	Codice di PRINT	
4108	0	Marcatore di fine linea	
4109 = =	19	Low Link	- Indicano l'indirizzo 4115 ++
4110	16	High Link	
4111	30	Low numero di linea	
4112	0	High numero di linea	
4113	128	Codice di END	
4114	0	Marcatore di fine linea	
4115 ++	0	Low link	- Essendo pari a 0 indicano la fine del testo.
4116	0	High link	

SIMULAZIONE DI UN PROCESSO D'APPRENDIMENTO

Sono ormai in molti i possessori di un micro-computer annoiati dal classico "MASTER MIND", un game in cui l'essere umano si prodiga nel tentativo di indovinare un numero ideato dal computer.

```
1000 REM"────────────────────────────────
1001 REM"  MASTER MIND: IL COMPUTER
1002 REM"INDOVINA IL NUMERO CHE PENSI!
1003 REM"   -- MAURIZIO DI VIZIO --
1004 REM"
1006 REM"   TEL 039/743092 . MONZA (MI)
1006 REM"
1010 REM"────────────────────────────────
1012 PRINTCHR$(14):REM  SET MINUSCOLE
1014 PRINT"███████████":A$="          "
1016 FORK=0TO39:PRINT"‾";:NEXT
1018 PRINTA$;"  ♥IMULAZIONE DI UN"
1022 PRINTA$;"      PROCESSO DI"
1022 PRINTA$;"     APPRENDIMENTO ."
1024 FORK=0TO39:PRINT"_";:NEXT
1026 FORK=0TO2000:NEXT
1028 DIM TEN(1000) , RTEN (1000)
1030 MAX = 124 :W=FRE(0)
1032 CICLO = 1
```

E' questa premessa la giustificazione razionale del programma presentato. Si tratta in effetti di un originale "gioco", se così può essere ancora definito, che stravolge la ormai esasperata e consumata idea vitale del "MASTER MIND".

In effetti spetta ora al "computer" il compito, apparentemente arduo, di precisare mediante successivi tentativi un numero pensato dall'utente.

Come funziona il tutto

In genere quando si vuole arrivare a dimostrare una tesi, si parte da presupposti, ipotesi. Nota comune di queste ultime è il fatto che nessuna di essa ne contraddice un'altra, ma tutte sono vere "contemporaneamente".

Nel caso specifico del programma, si tratta di procedere per tentativi alla "razionale" ricerca del numero misterioso.

L'aggettivo "razionale" si riferisce al fatto che il programma nella sua esecuzione tiene conto

```
1034 :
1036 REM------------------------------
1038 REM TENTATIVO STANDARD N. #1
1040 REM------------------------------
1042 :
1044 PRINT"■■■IL MIO NUMERO E' ■"123"■"
1046 INPUT"■■QUANTI PIU' = ";P
1048 INPUT"■■QUANTI MENO = ";M
1050 TEN (CICLO) = 123
1052 RTEN (CICLO) = (P+M/10)
1054 CICLO= CICLO + 1
1056 :
1058 REM------------------------------
1060 REM TENTATIVO STANDARD N. #2
1062 REM------------------------------
1064 :
1066 PRINT"■■■IL MIO NUMERO E' ■"456"■"
1068 INPUT"■■QUANTI PIU' = ";P
1070 INPUT"■■QUANTI MENO = ";M
1072 TEN (CICLO) = 456
1074 RTEN (CICLO) = (P+M/10)
1076 CICLO= CICLO + 1
1078 :
1080 REM------------------------------
1082 REM TENTATIVO STANDARD N. #3
1084 REM------------------------------
1086 :
1088 PRINT"■■■IL MIO NUMERO E' ■"789"■"
1090 INPUT"■■QUANTI PIU' = ";P
1092 INPUT"■■QUANTI MENO = ";M
1094 TEN (CICLO) = 789
1096 RTEN (CICLO) = (P+M/10)
1098 REM     CICLO=CICLO+1
1100 FOR X= MAX TO 998
1102 PRINT"■■■■■■■■■■■■";
1104 PRINT"■■■■■■■■■■■";
1106 PRINT"■A T T E N D E R E"
1108 :
1110 REM------------------------------
1112 REM INIZIA IL CICLO DEI TENTATIVI
1114 REM------------------------------
1116 :
1118 FOR C= 1   TO CICLO
1120 RI =000
1122 A$=RIGHT$(STR$(X),3)
1124 A1=VAL(LEFT$(A$,1))
1126 B1=VAL(MID$(A$,2,1))
1128 C1=VAL(MID$(A$,3,1))
1130 A$=RIGHT$(STR$(TEN(C)),3)
1132 A2=VAL(LEFT$(A$,1))
1134 B2=VAL(MID$(A$,2,1))
```

di elementi che con l'aumentare delle prove, divengono le "ipotesi" vere e proprie nel processo di dimostrazione della nostra tesi (e cioè indovinare il numero misterioso dell'utente).

Nel dettaglio il funzionamento del programma può essere così sintetizzato: l'utente pensa un numero, il computer fa un tentativo al quale si deve rispondere secondo un codice che poi vedremo; il tutto si ripeterà sino a che la macchina individua le nostre cifre (in generale 5,6 tentativi sono più che sufficienti).

E' interessante notare che se tentiamo di rispondere alle richieste della macchina mediante informazioni volutamente inesatte, il computer si accorge subito dell'imbroglio, identificabile con l'esistenza di ipotesi, (dati cioè da noi inseriti) che risultano contraddittorie.

Come rispondere agli INPUT

Il programma mostra i successivi tentativi della macchina: noi dovremo rispondere agli INPUT in seguito ad un confronto tra il numero proposto e quello da noi pensato.

Il computer chiede quanti "più e meno" ha realizzato mediante l'ultimo tentativo: noi dovremo a questo punto segnare su un foglio quanto segue:
numero pensato (es:) = 473;
numero proposto dal computer

```
1136 C2=VAL(MID$(A$,3,1))
1138 :
1140 REM-------------------------------
1142 REM CONTROLLI SUI PRECEDENTI TENT.
1144 REM-------------------------------
1146 :
1148 IFA1=A2THEN RI=RI+1
1150 IFB1=B2THEN RI=RI+1
1152 IFC1=C2THEN RI=RI+1
1154 IF A1=B2 THEN RI =RI + .1
1156 IF A1=C2 THEN RI =RI + .1
1158 IF B1=A2 THEN RI =RI + .1
1160 IF B1=C2 THEN RI =RI + .1
1162 IF C1=A2 THEN RI =RI + .1
1164 IF C1=B2 THEN RI =RI + .1
1166 IF RI <> RTEN (C) THEN 1200
1168 NEXT C :MAX = X
1170 :
1172 REM-------------------------------
1174 REM PROPONE UN ULTERIORE NUMERO
1176 REM-------------------------------
1178 :
1180 PRINT"               ";
1182 PRINT"TENTATIVO NUMERO # ";CIC
1184 PRINT"IL MIO NUMERO E' ";X;""
1186 INPUT"QUANTI PIU' ";P
1188 INPUT"QUANTI MENO ";M
1190 IF P=3 THEN GOTO 1224
1192 CICLO=CICLO+1
1194 TEN(CICLO)= X
1196 RTEN(CICLO)= (P+M/10)
1198 GOTO 1100
1200 C=CICLO : NEXT X
1202 :
1204 REM-------------------------------
1206 REM TEST DI EVENT. IRREGOLARITA'
1208 REM-------------------------------
1210 :
1212 PRINT"HAI BARATO !!!!"
1214 END
1216 :
1218 REM-------------------------------
1220 REM NUMERO OK . FINE PROGRAMMA .
1222 REM-------------------------------
1224 :
1226 PRINT"IL NUMERO E'";X
1228 PRINT"HO INDOVINATO IL NUMERO"
1230 PRINT"CON SOLI"CICLO"TENTATIVI"
1232 PRINT:PRINT:PRINT

READY.
```

(es:) = 123.

Controlliamo ora se, del primo numero, la prima, la seconda, la terza cifra sono rispettivamente uguali a quelle del secondo numero: in tal caso per ogni uguaglianza segnare un più (+) sul foglio.

Successivamente controlliamo se la prima cifra del primo numero è uguale alla seconda o terza del secondo numero; così si procederà per le altre restanti due cifre; a questo punto scriviamo sul foglio tanti meno (-) quante sono le uguaglianze riscontrate, (non barate: come ho detto "LUI" se ne accorgerebbe).

Ora finalmente potete tranquillamente rispondere ai successivi INPUT.

Tengo a precisare che, come già affermato, in cinque o sei tentativi, il nostro computer dovrebbe individuare il numero misterioso.

E' bene tener presente che tale numero, cioè quello pensato dall'utente, deve essere compreso, per come è strutturato il programma tra un minimo di 123 ed un massimo di 987.

Concludo dicendo che tale programma non è assolutamente lungo, per cui risulta possibile adattarlo anche al Vic 20 con la minima espansione (3K). ▪

Maurizio di Vizio

GESTIONE DEGLI SPRITE

Note esplicative del programma per disegnare sprite pubblicato sul N° 5.

Riga 10 • La POKE 650,128 abilita l'autorepeat su tutti i tasti.

150-155 • Memorizzazione delle routine in L.M. Fare molta attenzione nel copiare i data dalla riga 1000 alla 15500.

210 • Tasti da premere per muovere il cursore.

220 • Premendo il tasto "SHIFT" viene colorata la casella sotto il cursore. Premendo il tasto "Commodore" viene cancellata. E' possibile muovere il cursore tenendo premuto contemporaneamente lo "SHIFT" (o il "Commodore"), cioè colorare e cancellare muovendo.

Per memorizzare invece occorre che non sia premuto contemporaneamente alcun altro tasto.

600 • La routine in L.M. serve ad inizializzare le routine di trasformazione.

610 • Routine che trasforma l'immagine in cifre e la memorizza. Viene utilizzato il L.M. (Sys 49310) per velocizzare. Con il Basic occorrerebbero circa 20 secondi.

630-650 • Lo sprite viene visualizzato nei 4 formati possibili: n o r m a l e, espanso in altezza, espanso in larghezza ed espanso in entrambe le direzioni.

1010 • Routine in L.M. che riporta sullo schermo l'immagine memorizzata.

3000-3700 • Routine che si occupa di riempire le linee di data da 20000 a 21400.

3750 • Calcola il nuovo indirizzo di partenza dei programmi in basic.

Importante

4100 • Sposta puntatori dei programmi in Basic al nuovo indirizzo. Listando ora apparirebbero solo le linee da 20000 in poi. Per resettare: **POKE 43,1 - POKE 44,8.**

• Prima di salvare premere "Run/Stop" e "Restore" contemporaneamente, altrimenti il salvataggio non viene effettuato regolarmente (e ricordate di verificare);

• l'ultima figura memorizzata non viene persa finchè non si spegne il computer (resiste anche al reset del sistema (Sys 64738)) e il programma ne prevede il riutilizzo.

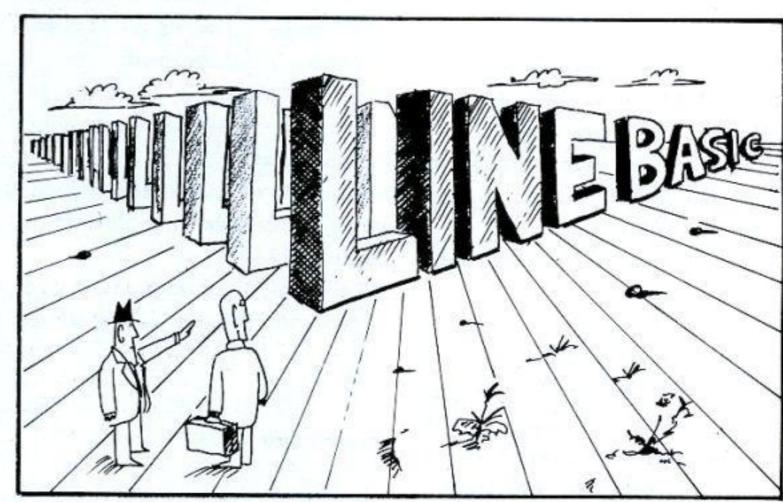
• se non avete scelto di salvare 4 copie sfasate di 90 gradi, alcune linee di data rimarranno vuote. Se non volete sprecare nastro e tempo potete cancellarle prima di salvarle. Badate, però, se intendete continuare ad utilizzare il programma, di reinserirle *esattamente uguali in lunghezza* o, se preferite, ricaricate il programma;

• le righe di data vuote sono tutte uguali in lunghezza, ed è molto importante batterle esattamente. Basta digitare il numero di linea (sempre di 5 cifre), lasciare uno spazio, poi battere "Data" e i "due punti", tornate col cursore sui "due punti" e premete "Shift" e "Insert/Delete". I "due punti" si fermeranno automaticamente quando saranno giunti all'ultima casella disponibile per la linea d'istruzioni. Premete "Return" ed il gioco è fatto;

• se volete inserire gli Sprite così creati o qualunque altro programma in un programma che avete già in memoria, senza distruggerlo, battete Poke 43, Peek (45)-2, Poke 44, Peek (46) e dopo aver caricato Poke 43,1: Poke 44,8. Attenzione che le righe del nuovo programma abbiano numeri di linea diversi da quelli del programma già presente. ▪

COME ALLUNGARE LE LINEE BASIC

Una limitazione, se così vogliamo chiamarla, dei calcolatori Commodore, è rappresentata dal numero massimo dei caratteri che possono essere digitati in ciascuna linea dei programmi basic: 80 nei computer della serie 2000, 3000, 4000, 8000, Commodore 64, e 88 nel caso del Vic 20. In effetti, a pensarci bene, il numero dei caratteri è più elevato dato che si può ricorrere ad abbreviazioni di comandi o istruzioni. Ciò è dimostrato dal fatto che, listando un programma, il numero dei caratteri che viene effettivamente visualizzato supera i limiti accennati. Se, per esempio, nel digitare una riga abbiamo usato il carattere "? ", nel listare la riga stessa esso viene "tradotto" con PRINT occupando, appunto, cinque caratteri e costringendo, in alcuni casi, a far apparire linee più lunghe del consueto.

In alcuni casi, ed in special modo ricorrendo ad istruzioni del tipo IF... THEN, il gruppo di comandi o istruzioni da soddisfare nel caso si verifichi un evento, può essere tanto consistente da costringere il programmatore a soluzioni che, pur se perfettamente valide, possono esser poco chiare per chi desidera studiare il listato. Ricorriamo ad un esempio chiarificatore: supponiamo che, a seconda del valore della variabile X, altre variabili debbano assumere particolari significati.

X=1:
A=1234.567 B=2346.567
C=5678,3455 A$="X UNITARIO"
D=1245.765 E=9886.345
F= 1360,4567 B$="PRIMO TENTATIVO"

X=2:
A=8654.345 B=7642.245
C=21467.466 A$="X DOPPIO"
D=7456.234 E=8904,006
F=23457.566 B$="SECONDO TENTATIVO"

X=3:
A=6543.345 B=3456.566
C=56788.234 A$="X TRIPLO"
D=9235.046 E=3467.567
F=13506.009 B$="TERZO TENTATIVO"
X=Eccetera.

Per soddisfare le condizioni indicate dovremmo ricorrere a istruzioni del tipo:

100 IF X=1 THEN A=1234.....
110 IF X=2 THEN A=8654...
Ecc.

Alcuni esempi

Prima della fusione.

```
90 .......
100 IF X=1 THEN A=1234.567: B=2346.567: C= 5678.3455: A$="X UNITARIO"
110 D=1245.765: E=9886.345: F=1360.4567: B$="PRIMO TENTATIVO"
120 IF X=2 THEN A=8654.345: B=7642.245: C= 21467.466: A$="X DOPPIO"
130 D=7456.234: E=8904.006: F=23457.566: B$="SECONDO TENTATIVO"
140 IF X=3 THEN A=6543.345: B=3456.566: C=56788.234: A$="X TRIPLO"
150 D=9235.046: E=3467.567: F=13506.009: B$="TERZO TENTATIVO"
160 .....

READY.
```

Dopo la fusione.

```
90 .......
100 IF X=1 THEN A=1234.567: B=2346.567: C= 5678.3455: A$="X UNITARIO"
    :::::D=1245 .765: E=9886.345: F=1360.4567: B$="PRIMO TENTATIVO"
120 IF X=2 THEN A=8654.345: B=7642.245: C= 21467.466: A$="X DOPPIO"
    :::::D=7456.2 34: E=8904.006: F=23457.566: B$="SECONDO TENTATIVO"
140 IF X=3 THEN A=6543.345: B=3456.566: C=56788.234: A$="X TRIPLO"
    :::::D=9235.04 6: E=3467.567: F=13506.009: B$="TERZO TENTATIVO"
160 .....

READY.
```

Sottoprogramma per fondere due linee successive.

```
62999 REM *** SUBROUTINE DI FUSIONE LINEE ***
63000 PRINT "LINEA DA FONDERE CON LA SUCCESSIVA "
63010 INPUT A: X = PEEK(43)+PEEK(44)*256
63020 X1 = PEEK(X): X2 = PEEK(X+1)
63030 I=X1+X2*256:J=PEEK(X+2)+PEEK(X+3)*256
63040 IF J=A THEN 63070
63050 IF J>A THEN PRINT A" = ■LINEA INESISTENTE ■" :END
63060 X=I : GOTO 63020: REM A. DE SIMONE
63070 X3=PEEK(I): X4=PEEK(I+1): POKEX,X3: POKEX+1,X4
63080 FOR J=-1 TO 3: POKE I+J, 58: NEXTJ

READY.
```

E evidente che il numero dei caratteri rappresentanti le istruzioni è nettamente superiore a quello accettabile dal computer. Come fare, allora in questi casi?

La soluzione più in uso è quella di ricorrere a subroutine da porre in fondo al programma. Esempio:

```
100 IF X=1 THENGOSUB1000
110 IF X=2  THENGOSUB1100
1000 A=1234.567: B= .....
C=. . . . : RETURN
1100 A=. . . . : B= . . . .
C= . . . . : RETURN
..........................................
..........................................
```

Eccetera

In questo modo, però, pur risolvendo il problema, dato che consente un numero praticamente illimitato di istruzioni in ciascuna subroutine, costringe il programmatore, in fase di esame del listato, a continui controlli del corretto indirizzo relativo alle varie subroutine richiamate. Indubbiamente sarebbe più interessante avere a disposizione, sulla stessa linea dell'IF... THEN tutti gli statement che la riguardano.

Un nuovo metodo

E veniamo, finalmente, alla descrizione del "trucco" che consente di allungare le linee basic.

Come è noto una linea basic è allocata, nella memoria RAM del computer, secondo uno schema ben rigoroso.

I primi due byte, detti di "link", indicano l'indirizzo della prossima linea del programma basic; i successivi due rappresentano la codifica del numero della linea stessa. Seguono i byte delle istruzioni contenute all'interno del programma. L'ultimo byte rappresenta *sempre* un valore nullo. Vediamo, in breve, come funziona il computer nel caso del semplice listato che segue:

```
100 PRINT "PROVA": PRINT
110 END
```

Quando battiamo RUN, il sistema operativo va ad eseguire il contenuto della prima linea allocata. Innanzitutto incontra due byte che gli comunicano il luogo della memoria in cui si trova la prossima linea basic; ne terrà conto quando, soddisfatti gli ordini impartiti nella linea appena incontrata, dovrà passare ad eseguire quelli che... seguono.

Dunque: incontra due byte di link; subito dopo viene informato che la linea che sta per trattare è numerata con 100 (sempre buono a sapersi). Il valore successivo è 153 che nel "gergo" Commodore significa PRINT. Esegue l'ordine visualizzando la parola: PROVA e capisce, dai due punti successivi, che c'è altro da fare: stampare un rigo a vuoto. Subito dopo incontra uno zero che viene interpretato come... fine della corsa. A questo punto, e *solo* adesso (dopo aver cioè incontrato uno zero) il sistema operativo va ad eseguire il contenuto della linea il cui indirizzo era indicato nei due byte di link visti prima (ricordate?).

Bene, il trucco è tutto lì. Se alla fine di una linea basic cancelliamo lo zero e, della linea successiva, sostituiamo i due byte di link e i due di numerazione basic con dei caratteri due punti (:) noi, ingannando il sistema operativo, lo costringiamo, a eseguire i vari ordini senza "uscire" dall'IF ... THEN. E' ovvio che, per completare l'opera, vanno alterati i due byte di link della linea basic che stiamo... allungando. Per essere più precisi, i due byte di link punteranno alla linea successiva a quella che vogliamo inglobare.

A tutte queste operazioni provvede la routinetta proposta da aggiungere in coda ai programmi che desideriamo alterare. Quando se ne ravvisa la necessità sarà sufficiente digirare:

RUN 63000

ed il sottoprogramma provvederà alla fusione della linea indicata con la successiva.

Gli "estratti" dei programmi pubblicati indicano il modo di digitare le linee da fondere e l'effetto che ne deriva dopo la fusione. ▪

Alessandro De Simone

FORMULA 1

Alla guida di una macchina bisogna riuscire a percorrere tutto il tracciato della pista senza urtare i bordi della pista stessa. Inizialmente verrà chiesto a quale livello di difficoltà si vuole giocare, verrà visualizzata la prima pista ed un semaforo che indicherà il momento della partenza. Dopo un po' di tempo la macchina comincia a muoversi. E' vietato tornare sui propri passi; in caso contrario il computer si accorge dell'imbroglio. Arrivati al traguardo compare automaticamente una nuova pista da percorrere, e arrivati al traguardo di questa, ne compare un'altra, e poi ancora, per un totale di cinque piste, messe in ordine crescente di difficoltà.

Se non riuscisse ad arrivare al traguardo della quinta pista, gli verranno assegnati dei punti in più ed una nuova macchina.

Si può uscire di pista, e quindi rompere la macchina solo due volte: alla terza il gioco termina.

Se si esce di pista e si dispone di alcune macchine, si riparte dall'inizio della pista nella quale si è verificato l'incidente.

Esistono poi dieci livelli di difficoltà. A livello zero la macchina va piano e non aumenta di velocità.

Negli altri livelli (dall'1 al 9), la macchina all'inizio andrà sempre piano, ma poi continua ad aumentare la sua velocità in proporzione al livello che si è scelto.

Il punteggio, che viene continuamente visualizzato nella parte alta del video, viene dato in base al livello di difficoltà e in base al numero di chilometri percorsi.

Al termine viene visualizzato il punteggio ottenuto. ▪

```
33 REM"
93 POKE53280,0:PRINT"▣":POKE53281,6
100 FORI=54272TO54272+28:POKEI,0:NEXT
101 POKE54296,15:POKE54277,0:H=54273
110 POKE54278,248:POKE54276,17
115 JP=56321:JN=255:JM=245:REM PER JOYSTICK
120 REM************************
130 REM**  F O R M U L A - 1 **
136 REM**                     **
133 REM**    VERSIONE C 64    **
140 REM**                     **
150 REM** DI GIOVANNI BELLU'  **
160 REM**                     **
170 REM**  VIA GIARDINI 20.   **
180 REM**                     **
190 REM**    SEREGNO  (MI)    **
200 REM**                     **
210 REM**  TEL. 0362/239580   **
220 REM************************
230 INPUT"▘QUALE LIVELLO DI DIFFICOLTA' (0-9)
    ▮▮▮*▮▮▮▮";DF:CP=DF
240 IFDF>9ORDF<0ORDF<>INT(DF)THEN230
250 ONDFGOTO260,270,280,290,300,310,
    320,330,340,350
260 LS=0:CF=2/3:GOTO370
270 LS=.1:GOTO370
280 LS=.3:GOTO370
290 LS=.5:GOTO370
```

```
300 LS=.8:GOTO370
310 LS=1:GOTO370
320 LS=1.5:GOTO370
330 LS=2:GOTO370
340 LS=2.5:GOTO370
350 LS=3:GOTO370
370 REM
420 PRINT"▟";
430 PRINT"     ______________POINT_________________________"
440 PRINT" /                 ▚ | O | ▒                    \"
450 PRINT"/                  ▚ | O | ▒                     \";
460 PRINT"| ˎ                ▚ | O | ▒                     |";
470 PRINT"|    /‾‾‾‾‾‾‾‾‾‾‾‾‾\_/‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾\      |";
480 GOSUB920
490 TI$="000000"
500 FORK=1TO10:GETA$:NEXT
510 POKEH,50:POKE1079,90:FORK=1TO500:NEXT:POKEH,100
520 POKE1079+80,90:FORK=1TO500:NEXT
530 POKEH,255:POKE1119,81:FORK=1TO50:NEXT:POKEH,0
540 POKE1079,87:POKE1159,87:POKE1119,87
550 B=42:A=1123:D1=32:C1=A:L=1:RM=255:V=25:L1=L
560 RM=RM-1:D=PEEK(A):IFD=127THEN780
570 IFTI$>T$THENPRINT"▟▙IMBROGLIONE CREDEVI DI FARMELA EH?!":POKEH,0:END
580 PT=PT+P+1:CC=CC+1:IFCC/5=INT(CC/5)THENV=V-LS
590 PRINT"▙"TAB(17)INT(PT*CP)
600 IFD<>32THENFORK=1TO10:GETA$:NEXT:GOTO1640
610 POKEC1,D1
620 D=PEEK(A):POKEA,B:POKEH,RM:POKEH+1,RM
630 FORK=1TOV:JK=PEEK(JP):IFJK<JMORJK=JNTHENNEXT:GOTO740
650 L1=L
660 IFJK=253THENL=40:GOTO740
670 IFJK=249THENL=39:GOTO740
680 IFJK=246THENL=-39:GOTO740
690 IFJK=245THENL=41:GOTO740
700 IFJK=250THENL=-41:GOTO740
710 IFJK=254THENL=-40:GOTO740
720 IFJK=251THENL=-1:GOTO740
730 IFJK=247THENL=1:GOTO740
735 NEXT
740 IFL=-L1THENPOKEH,0:PRINT"▟▙IMBROGLIONE SEI SQUALIFICATO !!!":END:END
750 D1=D:C1=A
760 A=A+L
770 GOTO560
780 REM POKEH,0:POKEH+1,0
790 P=P+1
800 POKEC1,D1
810 FORK=255TO0STEP-1:POKEH,K:NEXT:POKEH,0
820 PT=PT+100:PRINT"▙"TAB(17)INT(PT*CP)
830 IFP=5THENPT=PT+1000:FP=FP-1
840 IFP=6THENPT=PT+2000:FP=FP-1
850 IFFP=-1THENPT=PT+500:FP=0
```

```
860 IFP=5THENP=0:FP=0:PT=PT+5000:PRINT"S"TAB(17)INT(PT*CP):GOTO420
870 ONPGOSUB1400,1890,2120,2630
880 GOTO490
890 REM
900 REM PRIMA PISTA
910 REM
920 PRINT"\    |                             |  / ";
930 PRINT" |   |                             |  | ";
940 PRINT" |   |                             |  | ";
950 PRINT" |   |                             |  | ";
960 PRINT" |   |                             |  | ";
970 PRINT" |   \                             |  | ";
980 PRINT" |    \                            |  | ";
990 PRINT"  \    \                           |  | ";
1000 PRINT"   \    \                           |  |";
1010 PRINT"    \    \                          |  |";
1020 PRINT"     \    |                         |  |";
1030 PRINT"      |   |                        /   |";
1040 PRINT"      |   |                       /    |";
1050 PRINT"      |   |                      /    /"
1060 PRINT"      |   \_____________________/    /"
1070 PRINT"      \                             /"
1080 PRINT"       \                           /"
1090 PRINT"        \                         /"
1100 PRINT"         _________________________"
1110 PRINT"S"
1120 T$="000030":TI$="000000"
1130 RETURN
1380 REM SECONDA PISTA
1390 REM
1400 PRINT"SQQQQQ\  |                __________     |  / ";
1410 PRINT" |  |               /         \    |  | ";
1420 PRINT" |  \              /           \  /   | ";
1430 PRINT" |   \            /             \/    | ";
1440 PRINT" \    |          |   _____ \         / ";
1450 PRINT"  \   |          |   |      \       /  ";
1460 PRINT"   |  |          |   \_____  \     /   ";
1470 PRINT"   |  |          \          \ _____    ";
1480 PRINT"   |  |           \          \         ";
1490 PRINT"   |  |            \          \        ";
1500 PRINT"   |  \             ______ \   \       ";
1510 PRINT"   |   \ /____ \            \   \      ";
1520 PRINT"   \    \/      \           |    |     ";
1530 PRINT"    \            \          |    |     ";
1540 PRINT"     \            \________ /    /     ";
1550 PRINT"      \    /-\                  /      ";
1560 PRINT"       ____   \                /       ";
1570 PRINT"               \              /        ";
1580 PRINT"                \____________/         ";
1590 T$="000035"
1600 RETURN
```

```
1610 REM
1620 REM ROUTINE PER FUORI PISTA
1630 REM
1640 POKEC1,D1:PT=PT-10:PRINT""TAB(17)INT(PT*CP)
1650 POKEA,B:A1=PEEK(A-41):A2=PEEK(A-40):A3=PEEK(A-39):A4=PEEK(A-1)
1660 A5=PEEK(A+1):A6=PEEK(A+39):A7=PEEK(A+40):A8=PEEK(A+41)
1670 POKEA-41,77:POKEA-40,93:POKEA-39,78:POKEA-1,64:POKEA+1,64
1680 POKEA+39,78:POKEA+40,93:POKEA+41,77
1690 FORK=1TO50:POKEH,RND(1)*255:NEXT:POKEH,0
1700 POKEA-41,A1:POKEA-40,A2:POKEA-39,A3:POKEA-1,A4:POKEA+1,A5
1710 POKEA+39,A6:POKEA+40,A7:POKEA+41,A8
1720 POKEA,D
1730 FP=FP+1:IFFP=3THENPRINT"  SEI ANDATO FUORI PISTA TROPPE VOLTE ! "
1740 IFFP=3THENGOTO1770
1750 FORK=1TO30:GETA$:NEXT
1760 GOTO490
1770 FORK=1TO2000:NEXT
1780 IFDF<>0THENPT=PT*DF:GOTO1800
1790 PT=INT(PT*2/3)
1800 PRINT""
1810 PRINT" IL TUO PUNTEGGIO E' ";PT
1820 PRINT" VUOI GIOCARE ANCORA ?"
1830 GETA$:IFA$<>"S"ANDA$<>"N"THEN1830
1840 IFA$="N"THENEND
1850 RUN
1860 REM
1870 REM TERZA PISTA
1880 REM
1890 PRINT"\    \       ____________        /   /";
1900 PRINT" \    \    /       \         /   /  ";
1910 PRINT"  \    \  /         \       /   /   ";
1920 PRINT"   \    \/  /‾‾‾‾‾\  /     /   /    ";
1930 PRINT"    \   \/  /     /  /    /   /     ";
1940 PRINT"     \     /     /  /     \   \     ";
1950 PRINT"      \   /     /  //\     \   \    ";
1960 PRINT"       \_/     /  //  \    /   /    ";
1970 PRINT"   ___________/  //    \  /   /     ";
1980 PRINT"                //      \ |   |      ";
1990 PRINT"               //   /\   \/   /      ";
2000 PRINT"              //   /  \      /       ";
2010 PRINT"     /‾‾‾‾‾‾‾/    /    \    |        ";
2020 PRINT"     \______/    /      \  /         ";
2030 PRINT"                /          ‾‾‾         ";
2040 PRINT"             OIL  /                    ";
2050 PRINT"                /                    ";
2060 PRINT"               /                       ";
2070 PRINT"     ___________                     ";
2080 RETURN
2090 REM
2100 REM QUARTA PISTA
2110 REM
```

```
2120 PRINT"■■■■■■\  \              ____      /   /";
2130 PRINT" \  \             /    \   /  / ";
2140 PRINT"  \  |           |  /\  \ /  /";
2150 PRINT"   \ \/\____      |  | / /\  \  ";
2160 PRINT"    |       \    /  / | / \  \ ";
2170 PRINT"    \___/\   |  ___/  | | \   \ ■■";
2180 PRINT"        /  / /        / \  \  ■■ ■ ■";
2190 PRINT"       /  /  /  ■OIL■  |  \ \  ■ ■";
2200 PRINT"      |  | /       /      \ \ / / ";
2210 PRINT"      |  |/ /_____      |  | \  \ ";
2220 PRINT"      / / \   \        / /   \  \";
2230 PRINT"     / /    \   \      / /   /  /";
2240 PRINT"     \ \     \   \    / /  /  / ";
2250 PRINT"      \ \   / ■ ■|    |  | / /   ";
2260 PRINT"       \  \ | ■■ ■/   |  | / |   ";
2270 PRINT"        \  \/  /       \ \ |  |  ";
2280 PRINT"         \   /          \/  /   ";
2290 PRINT"          \_/           \____/ ";
2300 PRINT"                                 ";
2310 RETURN
2320 PRINT"■■■ SEI PRONTO ? "
2330 GETA$:IFA$<>"S"THEN2330
2340 RETURN
```

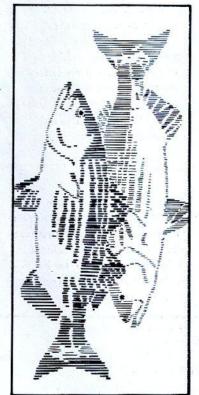
```
2350 FORK=1TO2000:NEXT:RETURN
2600 REM
2610 REM QUINTA PISTA
2620 REM
```

```
2630 PRINT"■■■■■■";
2640 PRINT"\  \              ___   __     /  /";
2650 PRINT" \  L__  ___     /  \ /  \    /  /";
2660 PRINT"  \    \/   \   | /\ \/ /\  \ /  /";
2670 PRINT"   ___\  /\  \   \ | \  \ / \  \  ";
2680 PRINT"       \  \   \   | \___  \ \ \  ";
2690 PRINT"       __/  \     \ \    \ \/ /  ";
2700 PRINT"      /      \     \ \    \___   ";
2710 PRINT"      | /___        \ \_____  \   ";
2720 PRINT"      |/             \      \  \  ";
2730 PRINT"      ||              ______\___  ";
2740 PRINT"      ||         ________    \  \ ";
2750 PRINT"      /|        /         \   ___\| ";
2760 PRINT"     / /    ___ | /_____\   /  ___/| ";
2770 PRINT"    | \   / \ | \      \,/   / /  ";
2780 PRINT"     \  \  \ |‾| \  \    \ \ |/____ ";
2790 PRINT"      \  \ | | | \ \  \   \ \ | \  ";
2800 PRINT"       \  \/ | \ \/  /    \  \/ /  ";
2810 PRINT"        \   /    \ /       \ /   ";
2820 PRINT"          ___                    ";
2830 TI$="000000":T$="000100"
2840 PRINT"■"
2850 RETURN

READY.
2860 REM REGOLO"_‾_‾_‾_‾_/_‾_‾_‾_‾_/_‾_‾_‾_‾_/_‾_‾_‾_

READY.
```

WARGAMES D'APRILE

Collegarsi col computer del Pentagono forse non è possibile ma...

Mario uscì dall'ascensore sovrappensiero e giunto alla porta dell'amico suonò il campanello. Subito avvertì un passo frettoloso, la porta che si apriva bruscamente e la voce eccitatissima dell'amico:
- Dai, entra, chiudi la porta e vieni in camera mia! - disse Franco mentre già si avviava in tutta fretta nella stanza.
- Ma che diavolo succede? - protestò il ragazzo che, lasciato il cappotto sulla sedia dell'ingresso seguì subito l'amico.

Nella stanza c'era un disordine, se era possibile, maggiore del s olito: dal personal computer fuoriuscivano alcuni cavi che terminavano nei pressi dell'apparecchio telefonico. Un oscilloscopio circondato da una caterva di nastri indicava chissà quale frequenza. Ma la cosa più inconsueta era proprio l'eccitazione di Franco, di solito impassibile anche di fronte ai più sofisticati programmi.
- Commodore Computer Club - si degnò di spiegare - ha pubblicato

il modo con cui mettersi in contatto con la rete computerizzata dei maggiori quotidiani italiani -
-... e con ciò? -
- Ma sei proprio deficiente! Non capisci che posso collegarmi col Corriere della Sera e avere sempre notizie dell'ultim'ora? -
- Non dirai sul serio! - esclamò Mario che incominciava a giustificare l'emozione dell'amico.
- Guarda tu stesso - ribattè Franco, e si mise a digitare come un pazzo.

Sul video del computer apparve la scritta:
- Ok: effettua il collegamento formando il numero 560 -
Franco sollevò la cornetta del ricevitore, attese il segnale di via libera e formò il numero indicato. Dopo alcuni istanti comparve il messaggio:
-Ok: collegamento effettuato, inserisci la parola d'ordine -
- Accidenti - esclamò Franco - Speriamo che sia la stessa di prima. Ho impiegato un'ora prima di scoprirla -.
Mario guardava ora il suo amico con un rispetto superiore al consueto: chi avrebbe mai detto che un ragazzino nè carne nè pesce come quello lì sarebbe riuscito a far tanto con un computerino che aveva sempre considerato un videogioco?
- Ma... - azzardò - non sarà mica proibito? -
- E chi se ne frega - fu la laconica risposta di Franco mentre, finalmente, il computer accettava la parola d'ordine digitata ed iniziava il collegamento.

- Diamine - disse Mario dopo un po' - ma sono proprio le notizie di oggi! - esclamò esaminando i titoli che a mano a mano apparivano sul video.

- Te l'avevo detto, no? - ribattè l'altro non facendo nulla per nascondere il proprio orgoglio. All'improvviso non apparve più alcuna notizia, lo schermo si sporcò in vari punti e, dopo un po' di tempo, ricomparve il messaggio: *componi il numero.*

- Ma che diavolo sarà successo? - esclamò Franco preoccupato per l'integrità del sistema. Digitò nervosamente alcuni tasti e si pose in attesa degli sviluppi. La scritta che apparve subito dopo fece rabbrividire i due ragazzi:

Base Nato: top secret
Confermata presenza missili nucleari Urss a 120 Km da Beirut puntato su Teheran. Controllo satelliti effettuato: positivo. Approntato sistema di intervento nucleare Nato di terzo grado...

A mano a mano che i caratteri apparivano sul video la bocca dei due giovani si apriva sempre di più, Mario sentì imperlarsi la fronte di sudore mentre esclamazioni soffocate venivano emesse di tanto in tanto senza che se ne rendessero conto.

B-52 con carico nucleare decollati da base 6-H68K. Confermato il decollo. Missili in Turchia allarme rosso. Il presidente Usa conferma l'attacco preventivo.

Una lunga pausa silenziosa permise ai due di guardarsi in faccia. Nuovamente il ticchettio riprese l'attenzione dei giovani!

Due missili della postazione TR98-ÜY sono stati lanciati.
e, dopo alcuni secondi:
Bersaglio centrato:
Buon pesce d'aprile! ! !

Franco guardò Mario, che ancora non capiva, e iniziò a ridere a crepapelle. Finalmente il ragazzo capì che era solo uno scherzo, si rimboccò le maniche ed iniziò a picchiare l'amico che si contorceva ancora dal ridere.

- Ora mi dirai che diavolo hai combinato altrimenti non smetti di prenderle -.

- Aspetta, fermati un po'... Non è altro che un programma che fa apparire una serie di messaggi l'uno dopo l'altro -.

- E il collegamento col telefono? - domandò Mario per nulla convinto della risposta.

- Ma che collegamento! Il filo elettrico che vedi è semplicemente attaccato con un po' di nastro adesivo sotto l'apparecchio: facevo solo finta di attendere la linea -

- E le notizie fresche di stampa? Quelle sono vere: le ho lette sul giornale di stamani -

- Anch'io le ho lette e non ho fatto altro che digitarle nei DATA del programma prima che tu arrivassi a casa mia -.

Mario si sedette ed iniziò a ridere anche lui. In fin dei conti lo scherzo era davvero riuscito, sul serio. ▪

Alessandro De Simone

```
100 REM ***********************
110 REM *    PESCE D'APRILE    *
120 REM *                      *
130 REM *  PER COMMODORE 64    *
140 REM *                      *
150 REM * DI GIOVANNI BELLU'   *
160 REM *                      *
170 REM * VIA GIARDINI N. 20   *
180 REM *                      *
190 REM *   S E R E G N O      *
200 REM *                      *
210 REM *  TEL. 0362/239580    *
220 REM *                      *
230 REM ***********************
240 POKE53280,0:REM BORDO NERO
250 PRINT"[CLR][RVS]PROGRAMMA PER IL COLLEGAMENTO CON ALTRI COMPUTERS"
```

VIC 20, ORA

SEMPRE A 199.000 LIRE!
PIÙ I.V.A.

CHE CE L'HAI...

Lo sai perchè VIC 20 va a ruba?
Provalo e scopri quante cose fa in più!
1. VIC 20 ha una valanga di videogiochi, uno più bello dell'altro, uno più nuovo dell'altro... e i successi dalle sale-giochi.
2. Ma VIC 20 è un vero computer, e fa molto molto di più.
3. Lo usi per la scuola, o per la casa, o per la professione. Trovi subito pronti un sacco di programmi.
Metti le cassette e via con cose utili. E nessun videogame ha i programmi del VIC.
4. VIC 20 ti serve per imparare il BASIC, la lingua del futuro (ed è facile imparare a programmare).
5. Nel mondo sono stati venduti più di un milione di VIC, a gente sveglia, quelli del 2000.
6. VIC 20 ha sempre un prezzo da sballo: costa solo 199.000 lire più IVA. E oggi lo trovi subito.

commodore
COMPUTER

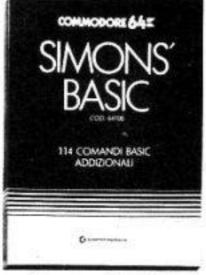
```
260 DIMN$(10,2)
270 FORK=1TO10:READN$(K,1):NEXT
280 FORK=1TO10:READN$(K,2):NEXT
290 S=54272:FORK=STOS+24:POKEK,0:NEXT:REM AZZERA
    IL CIRCUITO SONORO
300 POKE54296,15:REM VOLUME
310 A$="COLLEGAMI CON IL TELEFONO."
    :GOSUB740:GOSUB750
320 PRINT"OK -"
330 A$="CHIAMA IL NUMERO  560 ":GOSUB740
340 FORK=1TO15000:NEXT:REM 17 SECONDI DI ATTESA
350 A$=" OK - MI SONO COLLEGATO":GOSUB740
360 FORK=1TO5000:NEXT:REM RITARDO 5 SECONDI
370 A$=" INTRODUCI LA PAROLA D'ORDINE.":GOSUB740
    :PD=PD+1
380 PRINT""
390 INPUTA$:A=LEN(A$)
400 IFA=5THEN450
410 PRINT" NON VALIDA "
420 FORK=1TO1000:NEXT
430 PRINT"                                        "
440 GOTO390
450 PRINT" OK -"
460 A$="MI COLLEGO CON L'UTENTE PRESCELTO":GOSUB740
470 FORK=1TO5000:NEXT:REM RITARDO 5 SECONDI
480 IFPD=1THEN630
490 S=54272:FORK=STOS+24:POKEK,0:NEXT:REM AZZERA IL
    CIRCUITO SONORO
500 POKE54296,15:POKE54291,0:POKE54292,48:POKE54289,
    8:POKE54290,65
510 REM COLLEGAMENTO BASE NATO
520 PRINT""
530 A$=" BASE NATO : TOP SECRET@":GOSUB850
540 A$="————————————————————@":GOSUB850
550 PRINT:FORP=1TO10:PRINT""P" ":A$=N$(P,2)
    :GOSUB850:NEXT
560 FORK=1TO22:PRINT:NEXT
570 FORK=STOS+24:POKEK,0:NEXT
580 POKE54296,15:POKE54278,248:POKE54276,17
590 FORK=1TO5:POKE53280,K+2:POKE53281,K
600 FORJ=1TO255STEP2:POKE54273,J:NEXT
610 FORJ=255TO0STEP-2:POKE54273,J:NEXT
620 NEXT:POKE53280,0:POKE53281,0:PRINT"":END
630 A$=" CORRIERE DELLA SERA ":GOSUB740
640 A$="————————————————————":GOSUB740
650 PRINT
660 FORP=1TO10:PRINT""P" ":A$=N$(P,1):GOSUB740:NEXT
670 FORK=1TO5000:NEXT
680 POKES,220:POKES+1,68:POKES+5,15:POKES+6,215
690 POKES+7,120:POKES+8,100:POKES+12,15:POKES+13,215:POKE54272,
    0:POKE53272,0
```

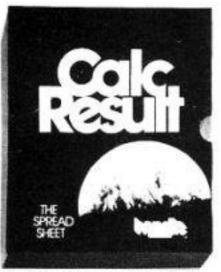
COMMODORE 64, ORA CHE CE L'HAI...

GUARDA CHE CI FAI

SCEGLI TRA CENTO E CENTO PROGRAMMI.

Corso di programmazione.

Il linguaggio didattico più divertente.

Per accedere a tutta la potenza del 64.

Linguaggio di programmazione della IVª generazione.

Pianificazione manageriale.

Facile registrazione dati.

Punta un dito e Commodore fa.

Per creare tutti i testi che vuoi. (con manuale)

24 ore da brivido.

Direttamente dalle sale-giochi.

Calcio spettacolo.

Un'orchestra in punta di dita.

Fai, fai, fai. Commodore 64 ha tantissimi programmi pronti, giusti per te. È ideale per hobby, lavoro, professione, gioco. Con programmi applicativi per la contabilità, l'amministrazione, le vendite, le previsioni, il word-processing...

Oltre a questa vastissima biblioteca, Commodore 64 dispone di un'infinità di "strumenti" per aumentare le proprie capacità. Fra questi, Forth 64, linguaggio di programmazione della quarta generazione, molto potente; Master 64, che mette a disposizione il basic 4.0 ed amplia lo stesso basic; Tool 64 che amplia la potenzialità del basic 2.2; Simons' basic, per aggiungere oltre 100 comandi al basic; Pet Speed, compilatore basic che aumenta la velocità d'esecuzione dei programmi di circa 40 volte.

Commodore Italiana S.p.A.
Tel. (02) 618321

```
700 FORK=1TO150:GOSUB1140:GOSUB1150:NEXT:POKE54273,0:POKE53272,21
710 A$="▩ IL COLLEGAMENTO SI E' INTERROTTO":GOSUB740:FORK=
    1TO2000:NEXT
720 PRINT"▣":GOTO360
730 REM SUB PRINT SENZA SUONO
740 FORK=1TOLEN(A$):PRINTMID$(A$,K,1);:FORJ=1TO250:NEXT:NEXT
    :PRINT:RETURN
750 REM SU -FATTO ?-
760 PRINTTAB(17)"FATTO ?"
770 FORK=1TO50
780 GETA$:IFA$<>"S"THENNEXT:GOTO800
790 K=50:NEXT:RETURN
800 PRINTTAB(17)"▣▩FATTO ?▣"
810 FORK=1TO50
820 GETA$:IFA$<>"S"THENNEXT:GOTO760
830 K=50:NEXT:RETURN
840 REM SUB PRINT CON SUONO
850 FORL=1TOLEN(A$):P$=MID$(A$,L,1)
860 IFP$="_|"THENPRINT:GOTO910
870 POKE54276,0:IFP$<>"@"THEN890
880 POKE54277,10:POKE54276,17:PRINT:FORL1=1TO500:NEXT:GOTO910
890 POKE54277,7:POKE54273,150:POKE54276,129:PRINTP$;
900 FORL1=1TO30:NEXT
910 NEXT:RETURN
920 REM DATA NOTIZIE ODIERNE.
930 DATA"▩NOTIZIE VARIE■"
940 DATA"FIRMATO IL CONCORDATO"
950 DATA"SLITTA L'ACCORDO SUGLI EUROMISSILI."
960 DATA"ATTENTATO A MADRID: MOLTE LE VITTIME"
970 DATA"FATALE INCIDENTE STRADALE A GENOVA"
980 DATA"NUOVA MISSIONE IN MAGGIO PER LO SHUTTLE"
990 DATA"▩NOTIZIE ECONOMICHE■"
1000 DATA"SALE ANCORA IL DOLLARO"
1010 DATA"PER IL MOMENTO STABILE L'ORO"
1020 DATA"LIRA IN LEGGERA FLESSIONE"
1030 REM DATA NOTIZIE BASE NATO
1040 DATA"▩ SITUAZIONE SOTTO CONTROLLO■@"
1050 DATA"SCOPERTI DAL ZK54 NUOVI MISSILI URSS@"
1060 DATA"POSTAZIONE SOTTERRANEA -▩OK■@"
1070 DATA"MANDARE SPIA S1A54 A MOSCA@"
1080 DATA"CONTROLLO SATELLITI EFFETTUATO@"
1090 DATA"PROGETTO PQT54/2 ▩ RIUSCITO ■@"
1100 DATA"DATA STABILITA PER L'ESPLOSIONE:▩25/5/84@"
1110 DATA"SPIE INVIATE DAL PENTAGONO: ▩12■@"
1120 DATA"ASPETTARE DIRETTIVE DAL PRESIDENTE USA@"
1130 DATA"▩ SEI PROPRIO UN PESCE: CI SEI CASCATO @"
1140 POKES+4,129:POKES+4,128:RETURN
1150 POKES+11,129:POKES+11,128:RETURN
1160 REM"REGOLO‾_‾_‾_‾_/‾_‾_‾_‾_/‾_‾_‾_‾_/‾_‾_‾_‾_/

READY.
```

ROUTINES GRAFICHE IN LINGUAGGIO MACCHINA

Come tutti i possessori di Commodore 64 sanno, non esistono comandi del Basic implementato che gestiscano l'alta risoluzione e, di conseguenza, sfruttare le buone capacità grafiche della macchina è piuttosto complicato. Le routines in L.M. pubblicate risolvono alcuni dei problemi più importanti.

• *Abilitazione modo testo/modo grafico.* La routine parte dall'indirizzo decimale 49821. Se ciò che "vede" nella locazione 248 è 0, viene abilitata la pagina testo (cioè quella normale). Se invece il contenuto è diverso da 0 è abilitata la pagina grafica. Prima di lanciare la routine è pertanto necessario scrivere nella locazione 248 il valore adatto.

Esempio per abilitazione di pagina grafica: POKE 248,1: Sys 49821. Per tornare al testo: POKE 248,0: Sys 49821. Si può tornare al modo testo anche premendo "Run/Stop" e "Restore" contemporaneamente.

• *Colori della pagina grafica.* Come per la pagina normale, si hanno 2 colori da scegliere: uno per lo sfondo e uno per i punti "accesi". Per lo sfondo inserire il numero corrispondente al colore desiderato (da 0 a 15) nella locazione 252.

Per i punti è necessario, invece, riferirsi alla locazione 251 prima di eseguire Sys 49785.

Esempio: volendo i punti in bianco e lo sfondo in nero: POKE 251,1: POKE 252,0: Sys 49785. La nuova mappa colore è allocata dalla locazione 52224 al-

la locazione 53223.

• *Pulizia della pagina grafica.* Eseguendo Sys 49763, si ottiene la pulizia istantanea della pagina grafica, cioè la cancellazione di tutto ciò che era disegnato. La pagina grafica è collocata dalla locazione 57344 alla locazione 65343 dove, cioè, normalmente si trova il sistema operativo.

Ciò è reso possibile dalla modifica degli interrupt attuata dalle routines. Ponendo la pagina grafica in questa posizione non si sottrae spazio ai programmi basic. Le routines appena descritte possono essere eseguite sia direttamente che all'interno di un programma, mentre le routines seguenti devono necessariamente essere eseguite da programma.

• *Routines "Plot" e "Draw".* Prima di tutto, una breve spiegazione per chi non conoscesse il significato di questi due comandi. Plot X,Y significa: accendi il punto di coordinate X,Y.

Draw X_1, Y_1, X_2, Y_2 significa: traccia la linea di estremi X_1, Y_1 e X_2, Y_2.

Le 2 routines svolgono tali compiti velocemente e sono facili da usare dato che non richiedono istruzioni del tipo "POKE" per individuare le coordinate. L'unica POKE da fare è nella locazione 2 per far sapere alle routines se i punti (o le linee) devono essere accesi o spenti (cioè disegnati o cancellati). Se nella locazione 2 c'è 0, il punto (o la linea) definito dalle coordinate viene cancellato; se nella locazione 2 c'è un valore diverso da 0, il punto (o la linea) viene tracciato. Poichè il contenuto della locazione 2 resta immutato fino a che non si interviene con una POKE, non è necessario ripetere: POKE 2, X ogni volta che si usano le routines Draw e Plot.

Veniamo ora alle coordinate. Tutti coloro che si sono cimentati con gli sprites, sono a conoscenza delle complicazioni che sorgono quando l'ascissa supera il valore 255. Non è infatti possibile "pokare" valori superiori a 255 (o minori di 0) in una locazione di memoria. Con queste routines invece basta assegnare i valori delle coordinate a delle variabili "speciali". Esempio: X% = 300: Y% = 100: Sys "Plot".

Tali variabili sono speciali per due motivi. Il primo è che devono essere del tipo intero seguite cioè dal simbolo per cento (%). Ciò comporta una limitazione nei valori delle coordinate tra -32768 e +32767. Considerato che i valori che rientrano nello schermo vanno da 0 a 319 per le ascisse e da 0 a 199 per le ordinate non sorge nessun problema. Entro tali limiti potete assegnare qualunque valore, anche negativo: le routines si occupano del controllo delle coordinate e "Plottano" il punto solo se va tutto bene.

• *Il punto o la linea sono interne allo schermo.* Tale controllo viene effettuato dalla routine Draw su ogni punto della linea. Il vantaggio rispetto ad un controllo sui soli estremi è di poter tracciare anche le linee che hanno uno o entrambi gli estremi "out" (ovviamente viene disegnata la sola parte che rientra nello schermo), mentre con il controllo dei soli estremi, esse sarebbero rifiutate "in blocco".

Il rallentamento conseguente alla ripetizione dei controlli non è eccessivo: una linea da 300 punti viene tracciata in 7 centesimi di secondo.

• *La seconda condizione speciale* è che nella riga di partenza del

programma, o comunque, prima di ogni altra variabile, occorre definire le 2 (se si usa solo la "Plot") o 4 (se si usa anche la "Draw") variabili che conterranno i valori delle coordinate. Esempio: se il programma comincia dalla riga 10, tale riga dovrà essere del tipo: 10 A% = 0: B% = 0: C% = 0: D% = 0. Ovviamente i nomi delle variabili potete sceglierli diversi e non dovete obbligatoriamente assegnare loro valore 0. L'importante è definirle prima di qualunque altra variabile. C'è un'eccezione: potete inserirle in qualunque punto del programma se sono immediatamente precedute da un comando "CLR" (vedi linea 100 del listato). In ogni caso però sempre prima delle linee in cui si trovano le Sys "Draw" e "Plot". Il comando "CLR" cancella tutte le variabili presenti in memoria: quindi usatelo con attenzione.

L'ordine in cui vengono definite le variabili "speciali" nella 1ª riga è molto importante: la prima viene sempre considerata dalle routines come contenente il valore dell'ascissa del punto da plottare o, se è in esecuzione la "Draw", come depositaria dell'ascissa del punto di partenza della linea. La seconda ha le stesse funzioni solo che si tratta dell'ordinata. La terza viene considerata come valore dell'ascissa del secondo estremo della linea da tracciare. La quarta è l'ordinata dello stesso punto.

In altre parole, se prendiamo l'esempio precedente, facendo eseguire: A% = 50: B% = 100: Sys "Plot" si avrà lo stesso punto che si avrebbe con B% = 50 : A% =100: Sys "Plot" perchè avevamo stabilito che A% conteneva l'ascissa e B% l'ordinata.

Non è obbligatorio dare tutte le coordinate ogni volta che richiamate una routine: a quelle mancanti verrà assegnato automaticamente l'ultimo valore. Si guardino ad esempio le righe da 310 a 330. Le coordinate del punto di partenza delle linee (X1%; Y1%), vengono date una sola volta, poichè tale punto è lo stesso per tutte le linee da tracciare ed è quindi inutile ripeterne le coordinate ogni volta.

Se volete tracciare linee molto lunghe (molto più lunghe dello schermo), tenete presente che la routine impiegherà tanto più tempo quanto più lunga è la linea (qualche secondo per una linea da 30.000 punti) e in quel periodo di tempo il computer non accetta messaggi dalla tastiera ed inoltre l'orologio interno non viene incrementato (perchè gli interrupt sono disabilitati). Se la precisione del timer è per voi essenziale vi posso fornire una variante di queste routine che non ha questo inconveniente, ma sottrae spazio ai programmi in basic.

Per scrupolo devo avvertirvi che le linee da tracciare non devono essere più lunghe di 32767 punti altrimenti si avrà un overflow nelle operazioni di calcolo eseguite dalla routine e la linea che ne deriverà sarà "sballata". Esempio: se ponete X1% = -1500: X2% = 2000, quindi ΔX = 35.000, la linea che ne risulterà non sarà corretta.

Un ultimo importante avvertimento: se volete fermare un programma che sta eseguendo delle "Draw" o delle "Plot" fatelo esclusivamente con il solo tasto "Run/Stop". Quando sarete sicuri che il programma si è arrestato potrete premere anche "Restore". È' sempre a causa degli interrupt. Se premete "Restore" mentre una linea è in corso di tracciamento il computer si "inchioda" e dovete spegnerlo. A chi non ha mai visto il CBM 64 operare in alta risoluzione, vorrei ricordare che se i punti appaiono più o meno brillanti e di colori diversi da quanto desiderato, ciò è dovuto alla macchina e non alle routines. E' un difetto che scompare tracciando punti contigui.

Vorrei far notare una piccola "finezza". Se provate a far disegnare una linea di estremi (1,10) e (20,11) mediante l'istruzione "Plot" (esempio: FOR X = 1 TO 20: X% = X: Y% = 1/19.X: Sys "Plot": NEXT), il risultato sarà: 19 punti di ordinata 10 e l'ultimo di ordinata 11. Ciò è dovuto al fatto di considerare solo la parte intera delle coordinate, per cui anche quando l'ordinata assume valori prossimi a 11 viene plottato un punto di ordinata 10. "Esteticamente" il risultato che ne deriva è pessimo.
Con la routine "Draw", invece, si

UNICAZIONE AZIENDALE SI EVOLVONO.

SUPPORTI MAGNETICI 3M.
NZIONE AL PRIMATO TECNOLOGICO.

Il primo nastro per computer è stato prodotto dalla 3M nel 1952. Un primato che ha consentito la realizzazione dei supporti magnetici più affidabili e sicuri.

Le Diskette 3M, ad esempio. Omologate dai maggiori costruttori.

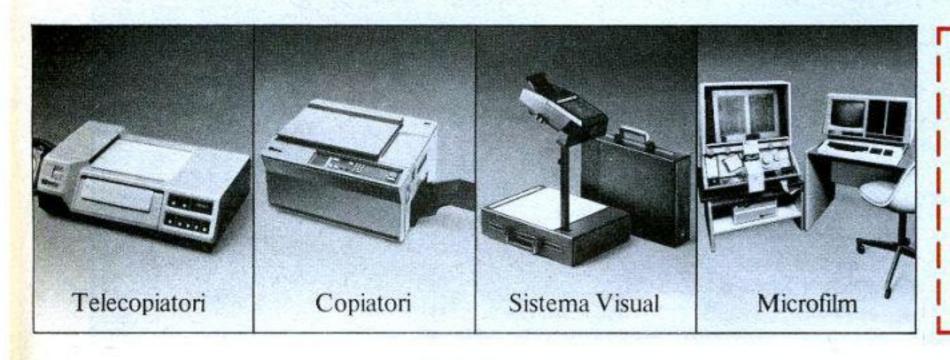
Certificate al 100%. Garantite 5 anni. Esportate in tutto il mondo. Prescelte come riferimento internazionale da ECMA, ISO ed ANSI. Una gamma completa sia per le Diskette 8 pollici che per le Minidiskette 5 pollici e un quarto.

Le Diskette 3M, grazie all'esclusivo rivestimento magnetico, garantiscono un'eccezionale resistenza all'usura e la massima affidabilità.

La stessa affidabilità che offrono tutti i Supporti Magnetici 3M: Nastri per Computer, Dischi Magnetici, Data Cartridge.

Le Diskette e i Nastri Magnetici 3M sono interamente fabbricati in Italia e questo significa immediata reperibilità e migliore assistenza.

Assistenza e consulenza tecnica che il Cliente trova presso le 8 filiali, i venditori diretti e 400 distributori 3M, in tutta Italia.

SISTEMI PER L'UFFICIO 3M. LA PERFEZIONE DELLA SPECIE.

Desidero ricevere ulteriori informazioni sui Supporti Magnetici 3M.

Nome ______

Azienda ______

Via ______

Cap ______ Città ______

C C C

Ritagliare e spedire a:
3M Italia S.p.A.-Linea Diretta-
Casella Postale 10411/10412-20110 Milano.
Oppure telefonare a: 02/75451

o t t i e n e una rappresentazione molto più naturale: 10 punti di ordinata 10 e 10 di ordinata 11; e questo senza che la routine preveda un controllo dei decimali (cosa che avrebbe complicato e rallentato l'esecuzione). Ciò avviene grazie ad un piccolo trucco nella determinazione del coefficiente angolare. Si suggerisce di trascrivere con molta attenzione il listato: un solo errore nella trascrizione dei "data" porterà quasi sicuramente all'inchiodamento del computer quando darete il "Run". Da prima salvate le routines su nastro (o su disco), altrimenti poi dovrete ricopiarle daccapo. Se invece non tutto è chiaro, o se gli esempi riportati nel listato non vi sono d'aiuto, potete telefonarmi.

Le routines vanno allocate a partire dalla locazione 49152 (vedi linea 70). Sono in una "zona" non accessibile dal basic e una volta immagazzinate non vengono più cancellate fino allo spegnimento della macchina. Non temono neppure il software-reset (Sys 64738). ▪

Ricordo gli indirizzi delle "Sys":

49693 per effettuare "Plot"
49728 per effettuare "Draw"
49763 per effettuare "Clear Screen"
49785 per effettuare "Color"
49821 per effettuare "Map/Text Mode".

Danilo Toma
Via Pordenone, 13 - Milano
Tel. 02/2159024

```
1 REM ** ROUTINES GRAFICHE PER CBM 64 **
2 REM ***  DI  DANILO  TOMA          ***
3 REM ***  VIA PORDENONE 13          ***
4 REM ***  MILANO TEL. 02/2159024    ***
5 REM ***                            ***
6 REM ********************************
7 REM
70 PRINT"▯"SPC(252)"▯LETTURA DEI DATA":FORI=49152TO49868:
   READA:POKEI,A:NEXT
90 REM
95 REM ***  ESEMPIO DI DRAW E PLOT  ***
99 REM
100 CLR:X1%=0:Y1%=0:X2%=0:Y2%=0:REM *** VARIABILI SCELTE
    PER LE COORDINATE
110 DR=49728:PL=49693:SC=49763:CL=49785:MO=49821:REM **
    INDIRIZZI DELLE ROUTINES
200 SYSSC:REM *** PULISCE LO SCHERMO
210 POKE251,3:POKE252,6:SYSCL:REM ***
    DEFINISCE I COLORI DELLA PAG. GRAFICA
220 POKE248,1:SYSMO:REM *** PASSA AL MAP MODE
300 POKE2,1:REM *** FLAG "DISEGNA/CANCELLA" SETTATO PER DISEGNARE
310 X1%=110:Y1%=100:FORI=0TO6.29STEP.1
320 X2%=SIN(I)*108+110:Y2%=COS(I)*100+100:SYSDR:REM *** DRAW
330 NEXT
340 FORI=0TO6.29STEP.1
350 X1%=SIN(I)*32+270:Y1%=I*32:SYSPL:REM *** PLOT
360 NEXT
380 IFPEEK(2)<>0THENPOKE2,0:GOTO310:REM *** FLAG SETTATO PER CANCELLARE
490 REM
500 REM ***  ALTRO ESEMPIO DI DRAW  ***
550 POKE2,1
560 FORI=0TO40:X1%=X2%:Y1%=Y2%
570 X2%=RND(1)*320:Y2%=RND(1)*200:SYSDR
580 FORK=0TO100:NEXT:REM *** CICLO DI RALLENTAMENTO
```

```
590 NEXT
600 FORK=0TO2000:NEXT
700 POKE248,0:SYSMO:REM *** PASSA AL TEXT MODE
999 END
3500 REM
3510 REM **** ROUTINES IN LINGUAGGIO MACCHINA  ****
3520 REM
3600 DATA 73,255,149,250,181,249,73,255,149,249,246,249,208,2,246
3700 DATA 250,160,202,162,255,96
5000 DATA 160,17,177,45,133,93,136,177,45,133,94,160,24,177,45,133,
     95,136,177
5010 DATA 45,133,96
5020 DATA 160,3,177,45,133,87,136,177
5030 DATA 45,133,88,160,10,177,45,133,91,133,78,136,177,45,133,
     92,133,79,96
5040 DATA 234,234
5100 DATA 160,232,162,0,56,165,93,229,87,133,249,165,94,229,88,
     133,250,16,3,32
5110 DATA 0,192,140,55,193,142,59,193,160,198,138,208,2,160,230,
     140,62,193
5200 DATA 160,232,162,0,56,165,95,229,91,133,253,165,96,229,92,
     133,254,16,5
5210 DATA 162,4,32,0,192,140,106,193,142,110,193,134,89,134,90
5215 DATA 160,198,138,208,2,160,230,140
5220 DATA 113,193,160,56,169,229,224,0,208,4,160,24,169,101,140,72,
     193,141,75
5230 DATA 193,141,81,193,141,87,193,141,93,193,169,234,141,115,193
5450 DATA 165,249,208,2,165,250,208,27,169,96,141,115,193,24,144,3
5500 DATA 32,104,193,32,141,193,165,78,197,95,208,244,165,79,197,96,
     208,248,96
5600 DATA 230,249,208,2,230,250,230,253,208,2,230,254
5650 DATA 169,0,133,251,133,252,133,247,133,248,162,33,165,247,56,229,249
5700 DATA 168,165,248,229,250,144,5,133,248,152,133,247,38,251,38,252,38
5800 DATA 253,38,254,38,247,38,248,202,208,224
5850 DATA 165,247,208,4,165,248,240,7,162,0,232,246,250,240,251
5900 DATA 160,1,165,253,208,6,165,254,208,2,160,0,132,80
6000 DATA 24,144,11,166,87,232,134,87,224,0,208,2,230,88
6050 DATA 165,91,133,78,165,92,133,79
6100 DATA 24,165,89,101,251,133,89,165,90,101,252,133,90,165,91,
     101,253,133,91
6130 DATA 165,92,101,254,133,92
6180 DATA 32,141,193,165,80,240,24,166,78,232,134,78,224,0
6200 DATA 208,2,230,79,234,165,78,197,91,208,231,165,79,197,92,208,248
6400 DATA 165,87,197,93,208,175,165,88,197,94,208,248,96
7050 DATA 165,88,240,12,201,1,240,1,96,165,87,201,64,144,1,96
7100 DATA 165,79,240,1,96,165,78,201,200,144,1,96
7250 DATA 165,78,41,7,133,247,165,78,74,74,41,254,168
7300 DATA 185,235,193,56,229,247,133,247,185,236,193,24,101,88,133,248
7400 DATA 165,87,41,248,168,165,87,41,7,170,189,227,193,166,2,240,5
7450 DATA 17,247,145,247,96,73,255,49,247,145,247,96
7500 DATA 128,64,32,16,8,4,2,1
```

$ 175.000 IN PALIO

Crea un videogioco intelligente ed originale e potrai diventare milionario. Questo fantastico concorso, organizzato dall'IRP (The International Register of Indipendent Computer Programmers ltd.) e dal famoso Marc McCormack International Management Group, e sponsorizzato per l'Italia da Commodore Computer Club, ti offre un'occasione unica nella vita. I premi sono elevati ed immediati, e ad essi si aggiungerà una royalty del 10% sulle vendite dei giochi premiati ai più importanti distributori in tutto il mondo. I vincitori, inoltre, potranno partecipare ad una serie di trasmissioni sulle principali reti televisive del mondo. La tua conoscenza dei computers e la tua immaginazione possono farti diventare rapidamente ricco e famoso!

PRIMO PREMIO: $ 100.000 *OLTRE*

CINQUE PREMI DI CONSOLAZIONE DI $ 15.000

Prepara un videogame nuovo e originale per una delle seguenti categorie: SPORT, SIMULAZIONI, ARCADE, STRATEGIA, AVVENTURA/FANTASIA o per la sezione speciale prevista per quei programmi che, pur non essendo dei veri giochi, abbiano notevoli caratteristiche didattiche o di divertimento. Sono altresì previsti una serie di premi al "Merito" che consentiranno di stampigliare sulle confezioni commerciali dei videogiochi selezionati la dizione "An International Videogame of The Year MERIT AWARD".
E' una grande sfida. Ed i premi, sia in termini finanziari che di prestigio, sono eccezionali.
Questo è certamente il concorso internazionale più eccitante per ogni appassionato di computer.

APPARIRETE IN TV!

I sei vincitori saranno invitati a far parte di uno spettacolare show televisivo che verrà distribuito ai più importanti network del mondo. Quanto basta per renderti famoso!

COME PARTECIPARE

Invia il tuo gioco o i tuoi giochi su cassetta indicando su quale computer gira o, girano, utilizzando questo coupon di qualificazione. Riceverai anche una documentazione completa con le regole dettagliate del concorso.

ENTRO E NON OLTRE IL 31 MAGGIO 1984

A: IRP Limited, Pinewood Film Studios, Iver, Bucks England

Nome: ....

Indirizzo ....

SC1

```
7600 DATA 7,254,139,252,135,251,71,250
7700 DATA 7,249,199,247,135,246,71,245
7800 DATA 7,244,199,242,135,241,71,240
7900 DATA 7,239,199,237,135,236,71,235
8000 DATA 7,234,199,232,135,231,71,230
8100 DATA 7,229,199,227,135,226,71,225
8200 DATA 7,224
8598 REM
8599 REM ****   PLOT   ****
8600 DATA 169,254,45,14,220,141,14,220,169,253,37,1,133,1
8700 DATA 32,43,192,32,141,193
8800 DATA 169,2,5,1,133,1,169,1,13,14,220,141,14,220,96
8999 REM
9000 REM ****   DRAW   ****
9100 DATA 169,254,45,14,220,141,14,220,169,253,37,1,133,1
9200 DATA 32,21,192,32,72,192
9300 DATA 169,2,5,1,133,1,169,1,13,14,220,141,14,220,96
58999 REM
59000 REM ***  PULISCE SCHERMO  ***
60000 DATA 169,224,133,250,169,0,133,249,162,32,168,145,249,200,
      208,251,230,250
60100 DATA 202,208,246,96
60148 REM
60149 REM ***  DEFINISCE COLORI  ***
60150 DATA 169,204,133,250,160,0,132,249,165,251,10,10,10,
      10,133,253,165,252
60200 DATA 41,15,24,101,253,162,4,145,249,200,208,251,230,
      250,202,208,246,96
60998 REM
60999 REM ***  TEXT MODE/MAP MODE  ***
61000 DATA 165,248,240,22,169,32,13,17,208,141,17,208
61050 DATA 169,56,141,24,208,169,252,45,0,221,141,0,221,96
61100 DATA 169,223,45,17,208,141,17,208
61150 DATA 169,21,141,24,208,169,3,13,0,221,141,0,221,96

READY.
```

```
10 REM    ESEMPIO DI APPLICAZIONE.
20 REM ESEGUIRE DOPO AVER ALLOCATO LE
30 REM ROUTINE IN LINGUAGGIO MACCHINA
40 :
90 REM DICHIARAZIONE QUATTRO VARIABILI
100 A1%=0:A2%=100:F1%=100:F2%=0
110 POKE248,1:REM MODO GRAFICO
112 SYS 49821:REM ABILITAZIONE ROUTINE
114 SYS 49763:REM CANCELLAZIONE SCHERMO
115 SYS 49728:REM PLOT LINEA
120 GET A$:IFA$=""THEN120:REM ATTESA
130 POKE248,0: SYS 49821:REM MODO TESTO

READY.
```

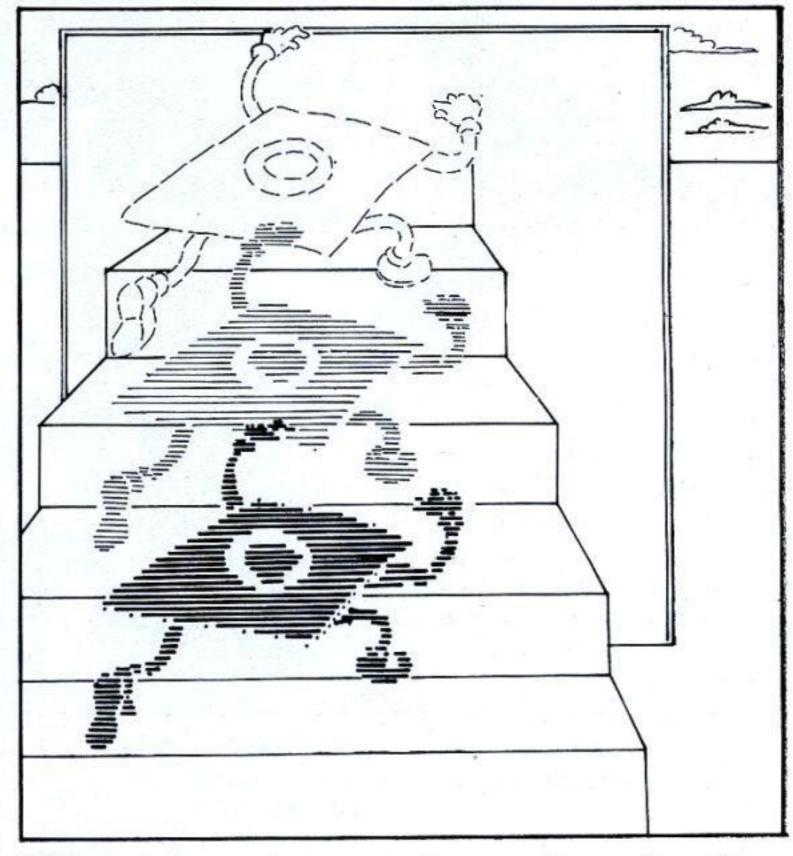
ARCHIVIO DISCHI

Il passaggio dal registratore al floppy, singolo o doppio, segna una svolta importante per chi vuole programmare: basta coi lunghi minuti di attesa per il caricamento dei programmi.

Quando però i dischetti cominciano ad aumentare di numero, non ci si ricorda più su quale disco sia stato registrato un certo programma. Ecco allora l'idea di rendere più agevole la ricerca, con l'uso di un programma appositamente studiato.

Esso infatti analizza il contenuto di ogni disco (vedremo in seguito come si fa), e lo memorizza... su un dischetto a parte.

Quando si vuole cercare un dato programma, sarà il computer a dirci su quale disco si trova. Questa, però, è solo una delle opzioni offerte dal programma, che, anche se piuttosto lungo, ricompensa con i vantaggi che offre la fatica di ricopiarlo, ma vediamo ora come si usa e quali sono le opzioni disponibili.

Prima di tutto bisogna registrare il programma "Archivio Dischi" su un dischetto nuovo, ma si può formattare un dischetto anche in modo diretto, digitando:

OPEN15,8,15, "NO: nome disco, ID"

Tale disco, insieme con il programma pubblicato nel presente fascicolo, farà da archivio per tutti gli altri dischetti.

Come si usa

Appena dato il RUN, bisogna dire se si dispone di un floppy singolo (1540 oppure 1541) o doppio, (4040 e simili) premendo S o D rispettivamente.

Quindi bisogna inserire nel Drive 0 il nostro disco-archivio e, quando tutto è pronto, si deve premere "S".

A questo punto il computer

leggerà alcuni dati dal dischetto (se è la prima volta che si usa questo programma, non leggerà niente, ma il programma funzionerà ugualmente) e, infine, mostrerà, una lista delle varie opzioni, che sono 5 oltre a quella di fine lavoro. Per scegliere l'opzione voluta è sufficiente premere il tasto numerico relativo ad essa.

Analizziamo ora ogni opzione.

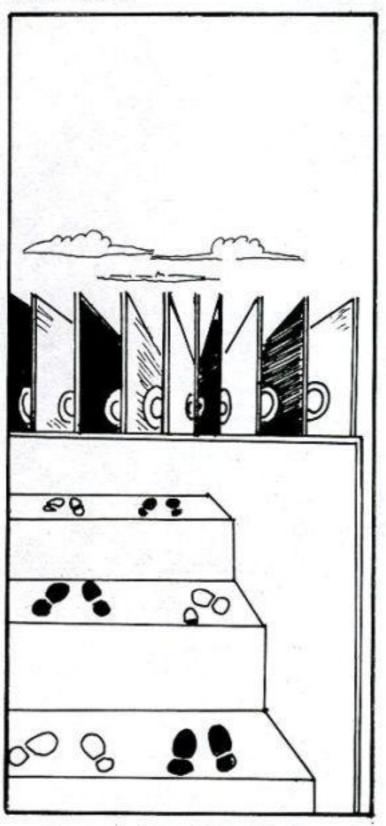
Analizzare un disco

Supponiamo ora, di avere un drive singolo. Dobbiamo quindi togliere il disco archivio ed introdurre quello da analizzare. Fatta questa operazione, premere un tasto qualsiasi. Premendo però "F" si ritorna alle opzioni principali. Se non abbiamo premuto "F", il computer leggerà il nome e l'ID del dischetto in esame, e chiederà se è proprio quello il floppy che si intende analizzare. In caso affermativo premiamo "S" (premendo invece "N" ritorna a chiedere di mettere il disco da analizzare nel drive 0).

A questo punto bisogna aspettare per permettere al computer di leggere e memorizzare la Directory, cioè i nomi di tutti i programmi registrati sul disco. Completata tale operazione, chiede di mettere il disco archivio nel drive 0, e, fatto ciò, di premere "S".

Comincerà quindi a memorizzare i dati ricavati dal dischetto precedentemente analizzato e a riordinare alcuni dati che gli serviranno successivamente in fase di ricerca. Alla fine torna a chiedere di mettere un altro disco da analizzare nel drive 0, e rispondendo "F" tornerà alle opzioni principali.

Cancellare i dati di un disco

Questo serve per cancellare dal disco archivio i dati relativi a un certo disco.

Bisogna dare l'ID del disco da cancellare, oppure, premendo Return, si può dare il nome, (premendo invece ancora Return si ritorna alle opzioni principali).

Supponiamo di voler cancellare i dati relativi al disco con ID "01": digitiamo allora "01" (Return).

A questo punto il computer visualizza il nome del disco e la sua ID e chiede se è il disco i cui dati intendiamo cancellare dall'archivio. Rispondendo "S" i da ti di quel disco verranno cancellati, e verrà richiesta ancora l'ID del disco da cancellare.

Premendo due volte Return, il computer riordina alcuni dati sul disco archivio e ritorna alle opzioni principali.

Output directory di un disco

isogna dare l'ID o il nome del disco (anche qui è valido lo stesso procedimento visto prima, per cui premendo due volte Return, si ritorna alle opzioni principali).

Se il disco richiesto è presente nell'archivio, se cioè è già stato fatto analizzare, ne verrà visualizzato il nome e l'ID. Se è quello giusto premiamo "S".

A questo punto vengono letti i dati di quel disco, precedentemente memorizzati con l'opzione 1, e verrà chiesto se si vuole mandare questi dati su stampante.

In caso contrario verranno visualizzati su video.

Durante la visualizzazione, premendo il tasto "F", la visualizzazione verrà interrotta e si ritorna all'inizio, altrimenti verranno visualizzati tutti i dati di quel disco.

Cercare un programma specifico

E' questa l'opzione più interessante, e più utile: ci permette di sapere su quali dischetti si trova un dato programma.

Bisogna dare il nome del pro-

RE LIBERO

gramma da cercare, e lo si può dare anche troncato, seguito dal simbolo "*". In questo caso verranno cercati tutti i programmi che iniziano con le lettere che precedono il simbolo "*". Per esempio, se si vogliono ricercare tutti i programmi che iniziano con "AR", bisogna digitare "AR*". Verranno visualizzati sia l'ID che il nome del disco su cui quel dato programma è registrato.

Alla fine, premendo un tasto, si ritorna all'inizio, e, premendo Return, alle opzioni principali.

List ID e nomi dischi

Ci sono, in questa opzione, altre cinque possibilità di scelta, più una per tornare alle opzioni principali.

• *Stampa mappa ID usate.* Si usa solo con la stampante.

Viene stampata una griglia sulla quale vengono indicate le ID già usate e quelle libere. Se si vuole interrompere la stampa, basta premere "F".

• *Veloce list delle ID usate.* Viene richiesto se si vuole mandare i dati su stampante o no, quindi vengono visualizzate tutte le ID usate nei dischi memorizzati, ed il numero totale dei dischi stessi.

• *List ID e nomi.* Anche qui si può scegliere se si vogliono mandare i dati su stampante, e poi vengono visualizzate sia le ID che i nomi dei dischi archiviati.

• *List min. blocks free per disco.* E' una ricerca dei dischi che hanno, come numero minimo di blocchi liberi, il valore che verrà introdotto. Per esempio, introducendo 100, verranno visualizzati tutti i dischi che hanno più di 100 blocchi liberi.

• *List max. blocks free.* Al contrario della opzione precedente, con questa vengono visualizzati i dischi che hanno, al massimo, il numero da noi introdotto di blocchi liberi. Per esempio, digitando il valore 99, vengono visualizzati tutti i dischetti che hanno meno di 99 blocchi liberi. ▪

Giovanni Bellù

```
100 REM **********************
110 REM ** ARCHIVIO DISCHI **
120 REM **                 **
130 REM **    A CURA DI    **
140 REM **                 **
150 REM ** BELLU' GIOVANNI **
160 REM **                 **
170 REM ** VIA GIARDINI 20 **
180 REM **                 **
190 REM **   SEREGNO (MI)  **
200 REM **                 **
210 REM ** TEL.0362/239580 **
220 REM **                 **
230 REM **********************
240 PRINT"▯HAI UN DRIVE ▯S▮INGOLO O ▯D▮OPPIO ?":CLR
250 GETW$:IFW$<>"S"ANDW$<>"D"THEN250
260 IFW$="D"THENDR$="1":GOTO280
270 DR$="0"
280 PRINT"▯METTI DISCO ARCHIVIO NEL ▯DRIVE 0▮ FATTO?"
290 GOSUB2420:IFC$<>"S"THEN290
300 C$="":S$=C$:X=0:Y=0:POKE59468,12
310 CR$=CHR$(13):HC$=CHR$(147):RV$=CHR$(18):RF$=CHR$(146)
    :CL$=CHR$(157)
320 REM
330 REM
```

```
340 GOSUB2440:DIMD$(230),X$(230)
350 PRINT"   STO LEGGENDO LA DIRECTORY PRINCIPALE"
360 OPEN15,8,15:PRINT#15,"I0":GOSUB2520
370 OPEN5,8,5,"0:DISK DIR XREF,S,R"
380 INPUT#15,EN,EM$,ET,ES:IFEN=62THEN490
390 INPUT#5,X$(NX):SS=ST:GOSUB2520:NX=NX+1:IFSS=0THEN390
400 GOTO490
410 S$="":FORX=1TOY:GOSUB450:S$=S$+C$:NEXTX:RETURN
420 FORX=1TOY:GOSUB450:NEXTX:RETURN
430 V=0:GOSUB450:IFC$<>""THENV=ASC(C$)
440 RETURN
450 GET#5,C$
460 SS=ST:INPUT#15,EN,EM$,ET,ES:IFEN=0THENRETURN
470 PRINTHC$;RV$;"ERRORE SU DISCO!":PRINT:PRINTEN;EM$;ET;ES
480 GOSUB2450:GOSUB2410
490 CLOSE4:CLOSE5:CX=0:GOSUB2440
500 PRINTSPC(5);"0 - FINE":PRINT
510 PRINTSPC(5);"1 - ANALIZZARE UN DISCO":PRINT
520 PRINTSPC(5);"2 - CANCELLARE I DATI DI UN DISCO":PRINT
530 PRINTSPC(5);"3 - OUTPUT DIRECTORY DI UN DISCO":PRINT
540 PRINTSPC(5);"4 - CERCARE UN PROGRAMMA SPECIFICO":PRINT
550 PRINTSPC(5);"5 - LIST ID E NOMI DEI DISCHI":PRINT
560 GOSUB2450:PRINT"SCEGLI L'OPZIONE DESIDERATA ";
570 GOSUB2420:IFC$="0"THENPRINTHC$:GOTO2680
580 V=VAL(C$):IFV<1ORV>5THEN570
590 ONVGOTO600,1460,1260,1980,1530
600 CX=0:PRINTHC$;"METTI DISCO DA ANALIZZARE NEL DRIVE "DR$
610 PRINT:PRINT"PREMI UN TASTO PER CONTINUARE  ":PRINTRV$;"
    F PER FINIRE"
620 GOSUB2420:IFC$="F"THEN490
630 GOSUB2450:PRINT"OK":PRINT#15,"I"+DR$:GOSUB460
640 OPEN5,8,5,"$"+DR$+",S,R":GOSUB460
650 GOSUB430:VF=V:GOSUB450:IFVF=67THENY=2:GOSUB420:GOTO680
660 NB=0:FORZ=1TO35:GOSUB430:IFZ<>18THENNB=NB+V
670 Y=3:GOSUB420:NEXTZ
680 Y=16:GOSUB410:DN$=S$:Y=2:GOSUB420
690 Y=2:GOSUB410:DI$=LEFT$(S$+"  ",2)
700 GOSUB2460:IFC$="N"THEN1250
710 IFNX=0THEN830
720 FORX=0TONX-1:C$=LEFT$(X$(X),2):IFDI$<C$THEN830
730 IFDI$<>C$THENNEXTX:GOTO830
740 IFDN$=MID$(X$(X),3)THEN830
750 PRINTHC$;RV$;"*** ATTENZIONE ***":PRINT" L'ID : ";
    RV$;DI$;RF$
760 PRINT:PRINT"DEL DISCO        : ";RV$;DN$
770 PRINT:PRINT"E' GIA' STATA CATALOGATA CON UN DIVERSO";
780 PRINT:PRINT"NOME DEL DISCO : ";RV$;MID$(X$(X),3):GOSUB2450
790 PRINT"SE CATALOGO QUESTO DISCO CANCELLO"
800 PRINT:PRINT"I DATI DEL DISCO ";MID$(X$(X),3):GOSUB2450
810 PRINT"DEVO CATALOGARLO ?";:GOSUB2490:IFC$="N"THEN1250
820 PRINTHC$;"STO CATALOGANDO IL DISCO CON UN NUOVO NOME"
```

Salva su dischetto questo listato. Il seguito verrà pubblicato nel prossimo fascicolo.

Guida mercato Commodore

Prodotto	Prezzo (IVA esclusa)

VIC - 20

Home Computer Vic 20 — 199.000
Unità di espansione *(1020)* — 295.000
Modulo di espansione *(1023)* — 135.000
Cartuccia da 3K di memoria *(1210)* — 49.000
Cartuccia da 8K di memoria *(1110)* — 75.000
Cartuccia da 16K di memoria *(1111)* — 125.000
Cartuccia Vic rel *(4011)* — 95.000
Permette di controllare il funzionamento di allarmi antifurto, porte automatiche, telefoni, trasmittenti ed apparecchi similari.
Vic switch *(4012)* — 225.000
Possono essere collegati fino a 16 VIC 20 con un floppy e una stampante (distanza massima 1500 mt.).
Interfaccia IEEE 488 *(T-1)* — 175.000
Interfaccia centronics *(T-3)* — 115.000
RS232-C adapter *(1011-A)* — 75.000
RS132-C adapter *(1011-B)* — 75.000

Commodore 64

CPU 64K RAM *(CBM64)* — 625.000
C 64 Executive *(SX 64)* — 2.350.000
Sistema operativo CP/M *(CP/M)* — 125.000
Consente di programmare il Commodore 64 in linguaggio CP/M, il più utilizzato sui Personal Computers. Permette inoltre di accedere alla enorme libreria di Software applicativi CP/M.
Pet speed *(6411)* — 95.000
Compilatore basic che aumenta la velocità di esecuzione dei programmi di circa 40 volte.

Accessori per Vic e Commodore 64

Stampante plotter a colori *(1520)* — 375.000
80 caratteri, per linea, 4 colori, alla risoluzione di 0,2 mm per passo.
Unità stampante *(MPS 801)*
Stampa velocemente su carta normale quanto appare sul video: programmi, lettere, dati, grafici.
Unità stampante *(1526)*
Stampante 80 colonne, bidirezionale, 60 CPS, spaziatura programmabile, trazione a frizione o a trattore.
Registratore dedicato *(1530)* — 120.000
Per memorizzare facilmente programmi e dati su normali cassette magnetiche.
Floppy disk drive *(1541)* — 630.000
Veloce unità di memoria di massa ad alta capacità. Può immagazzinare fino a 170.000 caratteri su ogni singolo disco.
Monitor monocromatico *(1601)*
A fosfori verdi 12".

Per giocare

Comando per giochi (Joystick) *(1311)* — 13.500
Permette di muoversi in tutte le direzioni, di iniziare i vari giochi di movimento e di "sparare".
Comando a manopola per giochi (Paddle) *(1312)* — 22.500
Adatto per i giochi a 2 persone, esegue movimenti in orizzontale e verticale,

Commodore 4000

CPU 16K RAM *(CBM 4016)* — 1.395.000
18K ROM, BASIC 4.0 residente, video 40 colonne per 25 righe, tastiera semigrafica.
CPU 32K RAM *(CBM 4032)* — 1.495.000
18K ROM, BASIC 4.0 residente, video 40 colonne per 25 righe, tastiera semigrafica.

Commodore 9000

Doppia CPU 134K RAM *(CBM 9000)* — 2.350.000
Micro Main Frame Computer a doppia CPU (6502 - 6809) compatibile con tutte le periferiche Commodore della serie 8000. Include 5 linguaggi di programmazione. (COBOL, FORTRAN, TCL PASCAL, UCSD PASCAL, APL).

Commodore 8000

CPU 32K RAM *(8032 SK)* — 1.725.000
18K ROM, Basic 4.0 residente, video orientabile e basculante 80 colonne per 25 righe, tastiera commerciale separata.
CPU 96K RAM *(8096 SK)* — 2.285.000
18K ROM, Basic 4.0 residente, video orientabile e basculante 80 colonne per 25 righe, tastiera commerciale separata. Include sistema operativo PM/96.

Commodore 600

Indicato per applicazioni industriali, collegamento a strumentazione, controllo numerico, ecc. Utilizza monitor in commercio.
CPU 128K RAM *(610)* — 2.150.000
CPU 128K RAM espandibileinternamentea 256K e esternamente a 960K, interfaccia RS232C, IEEE 488, porta utente a 8 Bit. Compatibile con tutte le periferiche Commodore della serie professionale.
CPU 256K RAM *(620)* — 2.550.000
CPU 256K RAM espandibili esternamente a 960K. Caratteristiche uguali al Mod. 610.
Monitor *(1601)* — 285.000
Monocromatico a fosfori verdi, 12".

Commodore 700

CPU 128K RAM *(710)* — 2.850.000
CPU 128K RAM espandibili internamente a 256K ed esternamente a 960K. Video orientabile e basculante 80 colonne per 25 righe. Compatibile con tutte le periferiche Commodore delle serie professionali.

CPU 256K RAM	*(720)*	3.250.000
Monitor a colori 14" con audio.	*(1702)*	645.000

Dischi

Floppy disk drive *(2031)* 675.000
Unità di memoria di massa ad alta velocità. Capacità 170KB. Drive singolo.

Floppy disk drive *(4040)* 2.095.000
Unità di memoria di massa ad alta velocità. Capacità 343KB. Drive doppio.

Floppy disk drive *(8050)* 2.375.000
Drive doppio 1M byte in linea.

Floppy disk drive *(SDF 1001)* 1.245.000
Drive singolo, doppia faccia, doppia densità 2M byte in linea.

Floppy disk drive *(8250 L.P.)* 2.600.000
Drive doppio, doppia faccia, doppia densità, 2M byte in linea.

Hard disk *(9060)* 6.900.000
Tecnologia Winchester, .5M byte in linea.

Hard disk *(9090)* 7.425.000
Tecnologia Winchester, 7.5M byte in linea.

Stampanti

Stampante *(4023 P)* 695.000
Bidirezionale ad aghi, 60 CPS, 80 colonne.

Stampante *(MPP 1361)* 1.275.000
Stampante ad aghi 150 CPS, 132 colonne, bidirezionale, trascinamento a trattore.

Stampante *(6400)* 3.250.000
Stampante a margherita, 40 CPS, 136 colonne passo pica, 163 colonne passo élite, bidirezionale, utilizzabile anche con carta da bollo, trascinamento a frizione o a trattore.

Accesori

Microprocessore 32K RAM *(MUPET II)* 2.500.000
Per connettere, in rete fino a 16 CPU RS232, IEEE 488, centronics. Il prezzo include (configurazione minima): controller, terminator, 3 moduli, cavi, cavo IEEE/PET (per la versione SK).

Singolo modulo addizionale: 325.000
1 modulo
1 cavo 6 piedi.

Nuovo sistema operativo *(PM 96)* 95.000
Per 8096SK o per 8032SK con B - 1 oppure con B - 2. Può gestire fino a 16 programmi residenti simultaneamente in memoria. Da a disposizione 26K per le variabili e 53K per i programmi. Potenzia inoltre il Basic con altri comandi.

64K RAM *(B-1)* 575.000
Scheda di ampliamento memoria per 8032 e nuovo sistema operativo "PM 96".

CP/Maker *(B-2)* 1.450.000
Incrementa la memoria interna di 64K RAM e permette l'uso di tutti i programmi CP/M. 8 bit disponibili. Compatibile con la serie 3000/4000/8000.

Scheda ad alta risoluzione grafica *(B-3)* 720.000
Compatibile ai sistemi della serie 8000.

Cavo PET/IEEE 488	*(C-1)*	85.000
Cavo IEEE 488/IEEE	*(C-2)*	95.000
Accoppiatore acustico	*(8010)*	595.000

300 baud/sec.

COMMODORE SHOP • L'UFFICIO 2000

VENDITA • ASSISTENZA TECNICA • APPLICAZIONI SPECIALI • PERMUTE E OCCASIONI

- *colmare la lacuna di disinformazione attorno al 64*
- *fornire uno strumento guida a tutti gli utilizzatori interessati*
- *mettere a disposizione di tutti i nostri 8 anni di esperienza Commodore*

SONO LE PRIME TRE FINALITA' CHE CI SIAMO PROPOSTI E PER LE QUALI ABBIAMO SCRITTO IL

C=64 LIBRO BLU C=64

COS'E?

Una raccolta seria di "cose" software e hardware esistenti sul 64.
Una illustrazione dettagliata della biblioteca programmi: UTILITY - DEMO - GESTIONALI - TECNICO SCIENTIFICO - DIDATTICI - VARIE.
Un approccio con l'hardware del 64 e relative periferiche per un corretto utilizzo e una prima manutenzione.

A COSA SERVE?

Ad orientarsi nel convulso mondo della microinformatica dove, mancando una informazione guidata ed organica spesso si sprecano tempo e soldi in vane ricerche.

A CHI SERVE?

È inutile dirlo. A tutti.
A chi possiede già un 64 o un qualsiasi computer Commodore.
E a chi pensa di acquistarne uno.

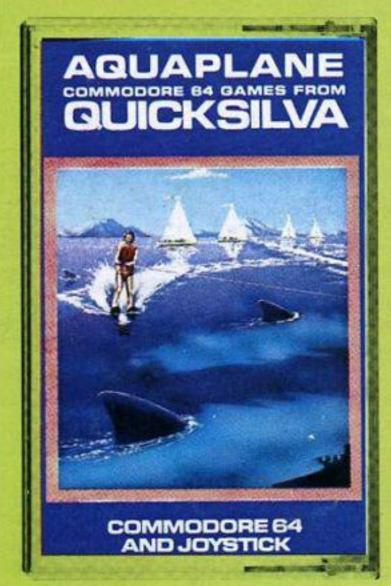
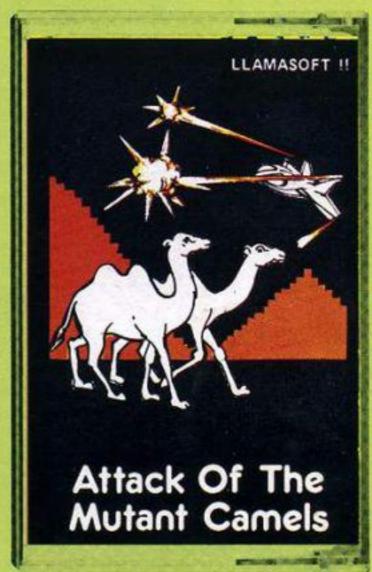
COME AVERLO?

Basta richiederlo per posta o per telefono.
Costa £ 25.000 e viene inviato pagando contrassegno.

LISTA OCCASIONI: KIT DIAGNOSTICO PER 8250 - TEST ALLINEAMENTO TESTINE PER FLOPPY 1541 - STAMPANTI AD AGHI E A MARGHERITA - DRIVES DA 5".

L'UFFICIO 2000 - Via Ripamonti, 213 - MILANO - Tel. 02/5696570 / 5696573 - **APERTO ANCHE DI SABATO**

Presso i Bit Shop Primavera

il software di casa...

ACQUAPLANE
Avreste mai pensato che praticare lo sci d'acqua potrebbe rivelarsi estremamente pericoloso? Jack Hollis, l'autore di Acquaplane pensa proprio di si. Il gioco, possiede tutte le caratteristiche per essere avvincente ed entusiasmante, la nostra abilità di sciatore sarà messa a dura prova.

ATTACK OF THE MUTANT CAMELS
Il pianeta terra ha bisogno di voi.
Gli alieni, esseri ostili, con l'impiego di una tecnica genetica, ha mutato i tradizionali cammelli, in portatori di morte.
Protetti da neutroni divenuti alti circa 3 metri. Potete manovrare i nostri bombardieri che volano a bassa quota sul terreno montagnoso, per distruggere i cammelli prima che essi invadono la roccaforte.

HEXPERT
Bert, la piccola e dolce creatura spaziale, che vive in lontano e sconosciuto pianeta e stata catturata da un malvagio diavolo chiamato Zoganaar.Questo tremendo mostro spaziale ha costretto Bert a partecipare ad un strano e mortale gioco: la costruzione di una piramide tridimensionale.
Riusciremo ad evitare al nostro amico Bert una brutta fine?

...sul vostro

commodore C64

naturalmente

SHOP bit primavera

COMPETENZA IN COMPUTER